

# Outlier Modeling in Image Matching

David Hasler, *Member, IEEE*, Luciano Sbaiz, Sabine Süssstrunk, and Martin Vetterli, *Fellow, IEEE*

**Abstract**—We address the question of how to characterize the outliers that may appear when matching two views of the same scene. The match is performed by comparing the difference of the two views at a pixel level aiming at a better registration of the images. When using digital photographs as input, we notice that an outlier is often a region that has been occluded, an object that suddenly appears in one of the images, or a region that undergoes an unexpected motion. By assuming that the error in pixel intensity generated by the outlier is similar to an error generated by comparing two random regions in the scene, we can build a model for the outliers based on the content of the two views. We illustrate our model by solving a pose estimation problem: the goal is to compute the camera motion between two views. The matching is expressed as a mixture of inliers versus outliers, and defines a function to minimize for improving the pose estimation. Our model has two benefits: First, it delivers a probability for each pixel to belong to the outliers. Second, our tests show that the method is substantially more robust than traditional robust estimators (*M*-estimators) used in image stitching applications, with only a slightly higher computational complexity.

**Index Terms**—Outlier model, outlier rejection, mixture model, robust pose estimation, *M*-estimators.

## 1 INTRODUCTION

IN image matching, a function finds corresponding pixels in two images. It enables, for example, to extract some information from the scene, to construct a 3D model, or to stitch a panorama. At some level of the computation, the matching function compares the pixel values of both images at matching positions, with the goal of iteratively refining the estimation. Even if there is a perfect geometrical match between the images, the pixel values will differ, either because of noise, or because there is something that violates the assumption made to do the match; in other words, an *outlier*. An outlier in an (photographic) image pair is often a region that has been occluded, an object that suddenly appears in one of the images, or a region that undergoes an unexpected motion. The fundamental problem of outlier rejection or robust estimation is to distinguish between noise, slight misregistrations, and outliers. In practice, given a difference between two pixel values, one should be able to decide whether, or to which extent, the pixel needs to be considered or discarded from the computation.

In this paper, we address this problem by extracting as much information as we can from the images prior to any matching computation. The key idea of this paper is to predict the outlier statistical characteristics as follows: We assume that, by comparing two arbitrary parts of each image, the resulting error pattern is similar to the one generated by an outlier. From this assumption, we can compute the expected error distribution generated by the outliers, which leads us to the *outlier model*. This model is

very general and can be applied to any image matching application. The second part of this paper shows how to apply the outlier model to a particular case of image matching, namely, to camera pose estimation.

The outlier model enables us to express a pose estimation problem as a *mixture of inlier versus outliers*, and to handle outlier rejection like a standard mixture problem [1], [2]. The performance of the model is demonstrated using two different types of experiments. The first experiment shows two aligned pictures, one of them containing an outlier. The proportion of the image covered by the outlier is varied by framing the images of the pair differently. The goal is to see when the model breaks down. The second experiment compares the pose estimator derived from our outlier model—which is an *M*-estimator—to a standard robust *M*-estimator.

After a brief discussion of the state-of-the-art in Section 2, the paper starts by presenting the outlier model in Section 3. In Section 4, we discuss how to apply the model to the pose estimation problem. We show two types of results. The first is how to discriminate inliers from outliers given a set of registered images. The second addresses the robustness of the pose estimation in the presence of an increasing number of outliers. In both cases, the model shows substantial improvement compared to existing techniques. Section 5 briefly discusses the limitations of the model, followed by the conclusion in Section 6.

## 2 STATE-OF-THE-ART

The state-of-the-art can be approached from two different points of view. On the one hand, there is general literature on robust estimation and outlier removal from the standpoint of statistics [3] that will not be further discussed here. On the other hand, there is the literature on robust *motion estimation*, for which [4, chapter 3] and [5] give an extensive review.

When handling images, outliers are detected by specifying a distribution for the inliers and using a threshold scheme if an observation diverges too much from the inlier data, as in [6],

- D. Hasler is with LOGO GmbH, 48565 Steinfurt, Germany. E-mail: david.hasler@bluewin.ch.
- L. Sbaiz is with Dartfish SA, 6, route del la fonderie C.P. 53, 1705 Fribourg 5, Switzerland. E-mail: sbaiz@dartfish.com.
- S. Süssstrunk and M. Vetterli are with the Audiovisual Communications Laboratory (LCAV), School of Computer and Communication Sciences (I&C), Swiss Federal Institute of Technology (EPFL), Ecublens INR-012, CH-1015 Lausanne, Switzerland. E-mail: {sabine.susstrunk, martin.vetterli}@epfl.ch.

Manuscript received 29 Oct. 2001; revised 16 June 2002; accepted 31 July 2002. Recommended for acceptance by D. Jacobs. For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 115286.

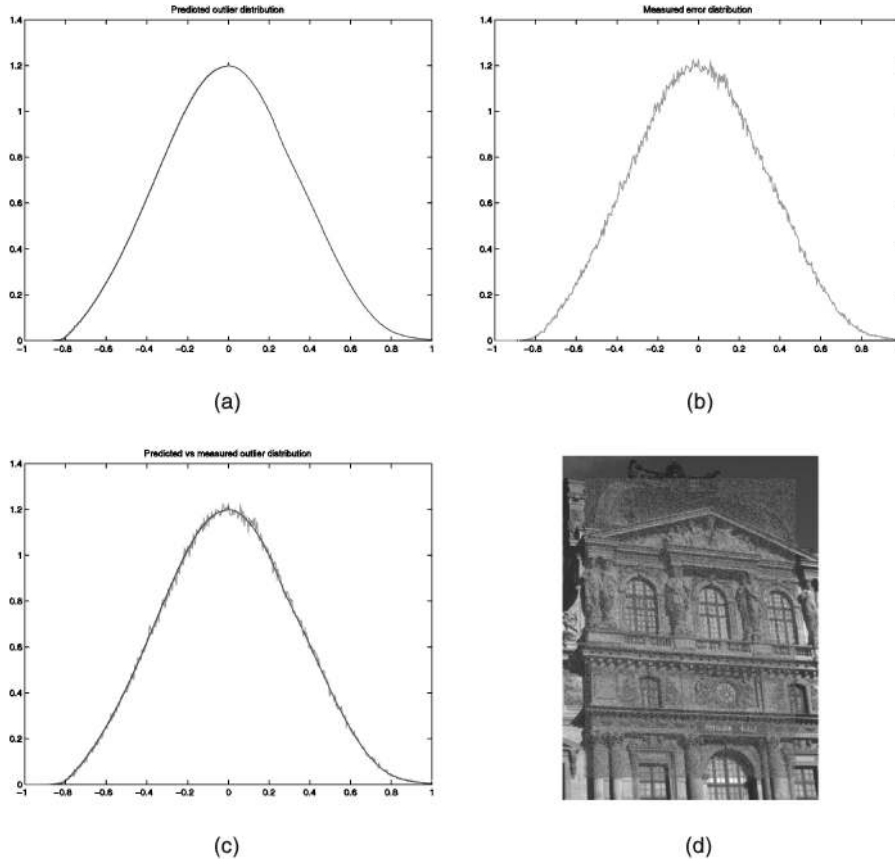


Fig. 1. Evaluation of the outlier model. (a) The predicted outlier distribution. (b) The error histogram. (c) Comparison of (a) and (b). The model and error histogram are superimposed and can hardly be distinguished. Therefore, the model works well in this case. (d) The image superposition used to generate the data: A white noise image has been compared to a real image (the images are rendered using a transparent blending).

[7], [8], [9]. More sophisticated methods [10], [11], [12] use a uniform distribution for the outliers and approach the estimation problem in the context of mixture modeling. In general, there is very little difference between the general statistical approach and the solutions that are applied to images.

Our approach follows a comparable principle than the work of Olson [14] who works with distances to edges rather than pixel-to-pixel differences, leading to substantially different results but sharing the same benefits. Gao and Boulton [15] also used a similar approach to subtract the background in a tracking problem, working at a pixel intensity level using a sophisticated time varying system.

### 3 THE OUTLIER MODEL

Considering a photographic image pair, we believe that *outliers* generate an error pattern similar to the error generated by comparing two random regions of the scene. Thus, we define the outliers as being the result of a random superposition of the images of a pair. The idea is to then characterize the outliers by computing their error distribution, with the result that we are able to handle any estimation problem as a mixture of inliers and outliers in a statistical framework.

Let  $I$  be the part of the scene that appears as  $I_0$  in image 0 and as  $I_1$  in image 1. We then compute the statistics of a random superposition of  $I_0$  and  $I_1$ . Let us

suppose that images 0 and 1 are the result of warping the scene  $I$  with two independent random variables  $\Theta_0$  and  $\Theta_1$  that are used as motion parameters:  $I_0 = I(\Theta_0)$  and  $I_1 = I(\Theta_1)$ . We are interested in the error distribution  $P(r) = \Pr\{I_0(\mathbf{p}) - I_1(\mathbf{p}') = r\}$ , where  $\mathbf{p}$  and  $\mathbf{p}'$  denote the matching positions in the images. Thus,

$$P(r) = \sum_{\forall u} \Pr\{I_0(\mathbf{p}) = u, I_1(\mathbf{p}') = u - r\}, \quad (1)$$

where  $u$  denotes all possible values contained in the images and  $r$  is the error value. Since  $\Theta_0$  and  $\Theta_1$  are independent, so are  $I_0(\mathbf{p})$  and  $I_1(\mathbf{p}')$ , thus

$$P(r) = \sum_{\forall u} \Pr\{I_0(\mathbf{p}) = u\} \Pr\{I_1(\mathbf{p}') = u - r\}. \quad (2)$$

By making some assumptions about the image and the comparison process (detailed in Appendix A), the intensity probability distribution of a single pixel in the image is equal to the image histogram  $\mathbf{H}$ , normalized such that  $\sum_u H(u) = 1$ . Equation (2) becomes

$$P(r) = \sum_{\forall u} H_0(u) H_1(u - r). \quad (3)$$

In other words, the outlier distribution is equal to the cross-correlation of the two image histograms, computed on the overlapping portion of the image pair. An experimental evaluation of (3) is presented in Figs. 1 and 2, verifying the

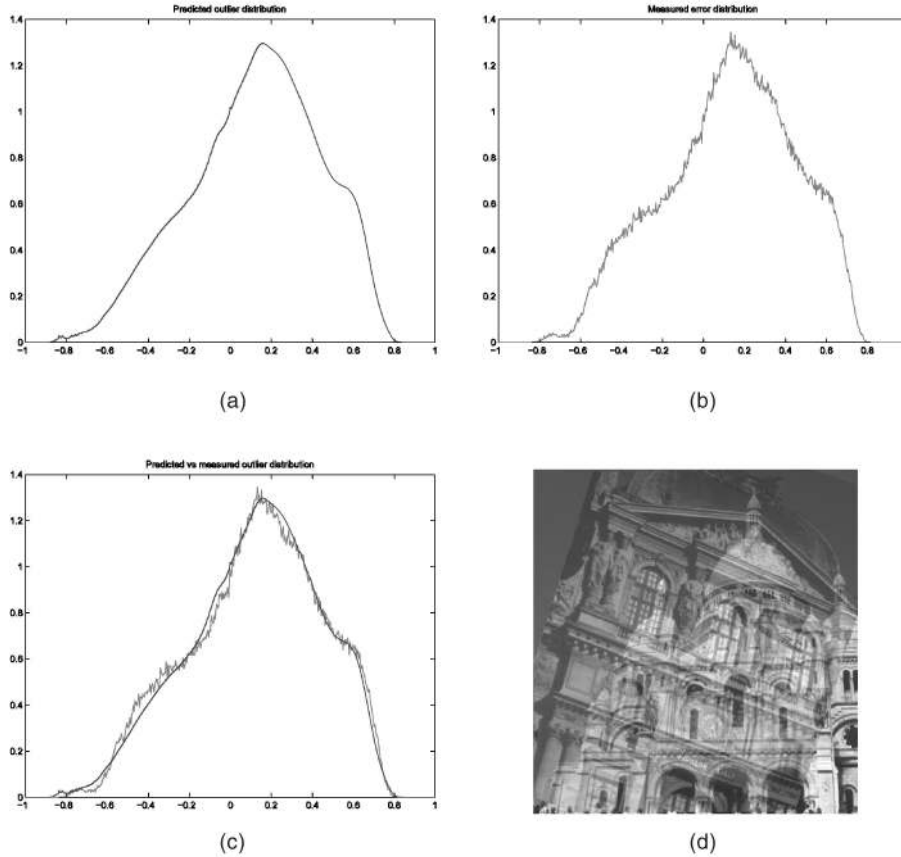


Fig. 2. Evaluation of the outlier modeling. (a) The predicted outlier distribution. (b) The error histogram. (c) Comparison of (a) and (b). We can see that the model explains the data fairly well. The histograms are computed without taking the sky in one of the pictures into account. (d) The image superposition used to generate the data (the images are rendered using a transparent blending).

validity of this model. The main assumption behind this result is that the expectation of the error histogram is equal to the error histogram delivered by a single image match. Further details can be found in Appendix A.

Note that this formulation does not take advantage of the correlation between neighboring pixels and assumes that the statistics of the part of the image that contains outliers is equal to the statistics of the whole image.

#### 4 APPLYING THE OUTLIER MODEL TO POSE ESTIMATION

Robust pose estimation consists in discounting the influence of the outliers when computing the pose. Standard robust pose estimation is performed using  $M$ -estimators which are reviewed in Section 4.1. In our model, we first need a model for the inliers (Section 4.2), to discriminate the outliers from the inliers. The two models are combined in the *OutlierMix* model in Section 4.3. This enables us to find the proportion of outliers (Section 4.4) and to compute for each pixel a probability to belong to the inliers (Section 4.5). Among existing methods, the most similar to ours uses a uniform distribution to model outliers and is presented in Section 4.3.1 for comparison purposes. The performance of the *OutlierMix* model to perform outlier detection is shown in Section 4.4.4 and its performance to estimate the pose of the camera is shown in Section 4.6.

##### 4.1 A Review of $M$ -ESTIMATORS

To discount the influence of outliers when matching two views of the same scene,  $M$ -estimators are applied along with an iteratively reweighted least squares (IRLS) approach [5]. In this regression, the  $M$ -estimator weights the influence of each pixel by a different factor. Formally, this is justified by assuming that the error distribution has a different shape than the usual normal distribution, like, for example, a Lorentzian or a Geman-McClur shape [4, chapter 3].

Let  $P_\sigma(\cdot)$  be the (discrete) assumed error distribution, which has one parameter:  $\theta$ . The likelihood of parameter  $\theta$  is defined by

$$L(\theta | P_\sigma) = \prod_r P_\sigma(r)^{n(r)}, \quad (4)$$

where  $n(r)$  is the number of occurrences of the error  $r$ . The maximum-likelihood value of  $\theta$  is found by solving

$$\hat{\theta} = \arg \max_{\theta} L(\theta | P_\sigma).$$

Since the logarithm is a monotonic and increasing function, the maximum likelihood of parameter  $\theta$  can be found by minimizing the negative log-likelihood

$$\hat{\theta} = \arg \min_{\theta} \sum_r n(r) (-\log[P_\sigma(r)]).$$

We call the negative log-likelihood the *objective function*  $\rho_\sigma$

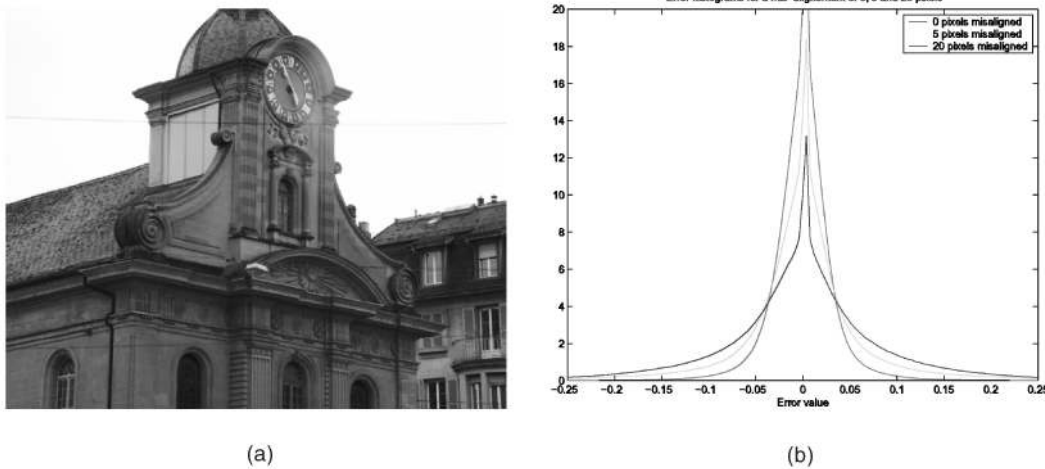


Fig. 3. Measuring inlier distributions. Two images looking like (a) are superimposed. (b) The error histogram for a perfect alignment, an alignment five pixels apart, and an alignment 20 pixels apart.

$$\rho_{\sigma}(r) = -\log[P_{\sigma}(r)] \triangleq \rho\left(\frac{r}{\sigma}\right). \quad (5)$$

To return to the pose estimation problem, finding the *Maximum-Likelihood* estimate for the given error distribution is equivalent to finding the minimum of the objective function

$$\hat{\theta} = \arg \min_{\theta} \sum_{i=1}^M \rho\left(\frac{r_i}{\sigma}\right), \quad (6)$$

where  $r_i$  is the error of a pixel-wise comparison of the images,  $\theta$  is the unknown parameter of the system (often a vector containing the motion parameters of the camera),  $M$  is the number of pixels being compared, and  $\sigma$  is the standard deviation of the error.

The most frequently used approach consists in assuming that the error distribution is Gaussian, which leads to the usual (nonrobust) minimum mean squared error estimate [13, chapter 11]. In addition to the Gaussian distribution, we will use two distributions: the Lorentzian and the Geman-McClur distribution. We choose the former for its precision of the estimation and the latter for its robustness to outliers, as we have found in our experimentation. The objective functions associated to these distributions (their negative log) is given, for the Gaussian<sup>1</sup> ( $\rho_N$ ), Lorentzian ( $\rho_L$ ), and the Geman-McClur ( $\rho_G$ ) distributions by

$$\rho_N\left(\frac{r}{\sigma}\right) = \frac{r^2}{2\sigma^2}, \quad (7)$$

$$\rho_L\left(\frac{r}{\sigma}\right) = \log\left(1 + \frac{r^2}{2\sigma^2}\right), \quad (8)$$

$$\rho_G\left(\frac{r}{\sigma}\right) = \frac{r^2/\sigma^2}{1 + r^2/\sigma^2}. \quad (9)$$

The weight  $W$  used by the  $M$ -estimators in the iteratively reweighted least-square iteration is given by  $W(r) = \frac{\dot{\rho}(r)}{r}$  [16, Appendix A], that is,

1. The  $N$  subscript stands for *Normal* distribution.

$$W_N(r) = \text{const}, \quad (10)$$

$$W_L(r) = \frac{2}{2\sigma^2 + r^2}, \quad (11)$$

$$W_G(r) = \frac{2\sigma^2}{(\sigma^2 + r^2)^2}, \quad (12)$$

for the Gaussian ( $W_N$ ), Lorentzian ( $W_L$ ), and Geman-McClur ( $W_G$ ) distributions, respectively. For an extensive discussion on standard robust estimation, we refer the reader to [4, chapter 3] and [5].

Note that, in order to use the Lorentzian or Geman-McClur estimator, we need to know the error scale  $\sigma$ , which is generally computed using the median of the error.

## 4.2 The Inlier Model

In an ideal situation, an inlier is an object that appears at the exact same location in the image pair, and has the exact same pixel value. In practice, the pixel values may differ because of acquisition noise, and the image might not be perfectly registered. Since the registration algorithm works in iterations, this slight misregistration is perfectly normal in the context of motion or pose estimation and it is crucial to consider the slightly displaced objects as inliers. Surely enough, these misregistered pixels are the ones that help the motion estimation algorithm achieve a better registration. This can be explained by the fact that iterative algorithms work on local relationships in images, and the slight displaced pixel should cause a decrease of the objective function if they get aligned in the next iteration.

In order to characterize the inliers, we conduct the following experiment: We take two images of a scene and make sure that nothing changes neither in the scene, nor in the camera pose and settings, and compute the error histogram—the difference in the two pictures is due solely to the acquisition noise. Then, we slightly misalign one of the pictures, and recompute the error histogram. We repeat the experiment for different misalignment amplitudes and try to fit a model to the error histogram. The error histograms of an image pair are shown in Fig. 3, for 0, 5, and 20 pixels of misalignment, respectively.

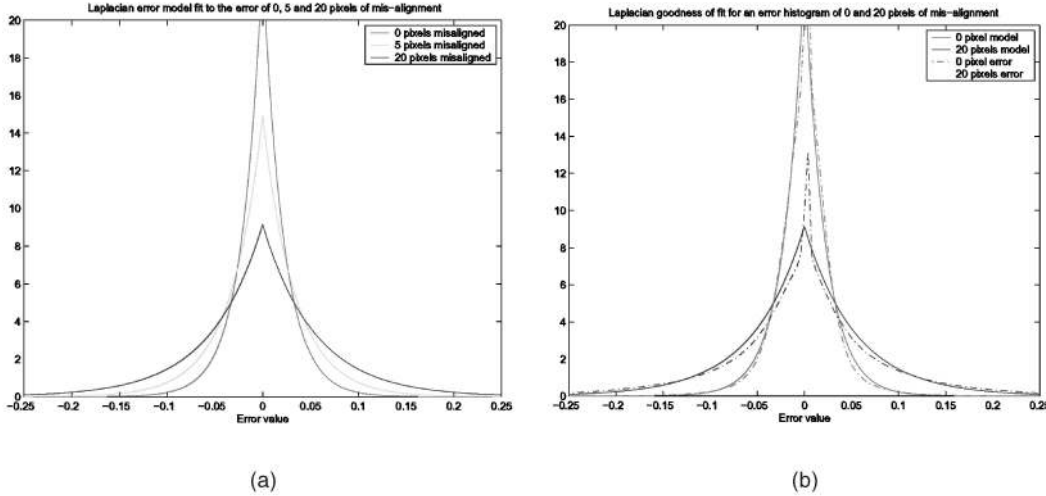


Fig. 4. Error distribution versus a Laplacian distribution. (a) Three different Laplacian distributions computed to match the error distribution of Fig. 3b. (b) Comparison of the Laplacian models in (a) with the error distribution in Fig. 3b (the histogram showing a misalignment of five pixels has been suppressed for clarity reasons).

Fig. 4 shows a comparison of the inlier histogram with a zero mean<sup>2</sup> Laplacian distribution. The Laplacian distribution ( $\mathcal{L}$ ) is defined by [17]

$$\mathcal{L}_{(0,\sigma)}(r) = \frac{1}{2\sigma} \cdot e^{-\frac{|r|}{\sigma}}, \quad (13)$$

where  $r$  is the residual (the registration error) and  $\sigma$  is the standard deviation. Although a generalized Gaussian distribution [18] would deliver a better fit, we disregarded this option because of the increased complexity that such a choice would bring. The generalized Gaussian distribution requires the estimation of two parameters instead of only one for the Laplacian. We therefore chose to use the Laplacian distribution to model the inliers, which requires us to find only its standard deviation  $\sigma$  that will differ with each estimation.

### 4.3 The OutlierMix Model

By combining the results of the two previous sections, we can build a new model for matching two images; we call it the *OutlierMix* model. This model is a mixture of the outlier model of Section 3 with the inlier model of Section 4.2.

Given the outlier distribution  $P_O$ , we can describe the error generated by matching two images as a mixture of inliers and outliers:

$$P(r) = \phi P_I(r) + (1 - \phi) P_O(r) \quad (14)$$

$$\mathbf{H}_m = \phi \mathbf{H}_I(\sigma) + (1 - \phi) \mathbf{H}_O, \quad (15)$$

where  $P_I$  stands for the inlier probability density,  $P_O$  is the outlier probability density,  $\phi$  is the proportion of inliers, and  $r$  the error value. Equation (15) is equivalent to (14), but using matrix notation, where  $\mathbf{H}_m$  denotes the model histogram,  $\mathbf{H}_I(\sigma)$  the inlier histogram, and  $\mathbf{H}_O$  the outlier histogram. As discussed in Section 4.2, we assume that the inlier distribution is a Laplacian distribution with zero mean:  $P_I \sim \mathcal{L}_{(0,\sigma)}$ . The model parameters  $\phi$  and  $\sigma$  are computed by fitting the model error distribution  $P(r)$  to the error histogram  $\mathbf{H}_e$ , according to Section 4.4.

2. If the settings of the camera are varying (even slightly), then the registration model should compensate for these changes to ensure that the inliers are zero mean.

Note that the outlier *distribution* depends only on the image histograms and not on the error. In other words, we can compute the outlier distribution without doing any image superpositions or any error computation. Nevertheless, to compute the outlier *proportion* (and the inlier standard deviation), we need to compute the error by superimposing the images.

#### 4.3.1 The UniformMix Model

A common practice is to model the outliers with a uniform distribution [10], [11], [12]. To compare this approach to the *OutlierMix* model, we introduce the *UniformMix* model, which is a mixture of the inlier model of Section 4.2 with a uniform distribution.

Given the outlier distribution  $P_O$ , we can describe the error generated by matching two images as a mixture of inliers and outliers:

$$\mathbf{H}_m = \phi \mathbf{H}_I(\sigma) + (1 - \phi) \mathbf{H}_U, \quad (16)$$

$$\mathbf{H}_U(r) \triangleq \frac{1}{N}, \quad \forall r, \quad (17)$$

where  $\mathbf{H}_m$  denotes the model histogram,  $\mathbf{H}_I(\sigma)$  is the inlier histogram,  $\mathbf{H}_U$  is the uniform distribution,  $\phi$  is the proportion of inliers, and  $N$  is the number of possible error values  $r$  that an image matching can produce. The model parameters  $\phi$  and  $\sigma$  are computed using the same procedure than for the *OutlierMix* model.

Note that this formulation differs from the one proposed in [12] by setting the inlier proportion  $\phi$  according to the image content.

### 4.4 Fitting the Model to the Error Distribution

Given the error distribution, the inlier model, and the outlier distribution, we can find the *inlier scale* ( $\sigma$ ) and the *inlier proportion* ( $\phi$ ) by fitting the model mixture to the measurements. The inlier scale is the standard deviation of the error generated by the inliers and is also called the *error scale*. Two methods are proposed: The first method computes the maximum likely parameters and the second method minimizes the  $L_2$ -norm between the *OutlierMix* model and the error distribution. This last choice is

motivated by computational complexity. The experiments are then repeated using the UniformMix model.

#### 4.4.1 Traditional Median Estimator

To compare the OutlierMix model to the standard  $M$ -estimators presented in Section 4.1, we compute the inlier scale using the median of the error value. The inlier scale can be derived from the median  $m$  by solving  $\int_0^m \mathcal{L}_{(0,\sigma)}(r)dr = 1/4$ , i.e.,  $\sigma = m/\ln(2)$ .

#### 4.4.2 Maximum-Likelihood Computation

The likelihood  $L$  (defined in (4)) of the inlier scale parameter  $\sigma$  and the inlier proportion  $\phi$  is given by [1, chapter 1]

$$L(\phi, \sigma | P_O) = \prod_r P(r)^{n(r)},$$

where  $P(r)$  is the OutlierMix model distribution given by (14),  $P_O$  is the outlier distribution, and  $n(r)$  is the number of occurrence of the error  $r$ . Using the histogram notation of Section 4.3, the likelihood can be rewritten as

$$L(\phi, \sigma | P_O) = \left[ \prod_r H_m(r)^{H_e(r)} \right]^\alpha,$$

where  $H_m(r)$  is the component at position  $r$  of the model histogram  $\mathbf{H}_m$ ,  $\mathbf{H}_e$  denotes the error histogram and  $\alpha$  a constant factor due to the normalization of  $\mathbf{H}_e$ . To find the maximum-likelihood estimate of the inlier scale  $\sigma$  and of the inlier proportion  $\phi$ , we maximize the log-likelihood function  $l(\cdot)$

$$\begin{aligned} l(\phi, \sigma | P_O) &\triangleq \alpha \cdot \log \left[ \prod_r H_m(r)^{H_e(r)} \right] \\ &= \alpha \cdot \sum_r H_e(r) \log[H_m(r)], \quad (18) \\ (\sigma, \phi) &= \arg \max \sum_r H_e(r) \log[H_m(r)], \end{aligned}$$

which is equivalent to maximizing the likelihood function. Equation (18) is solved numerically.

#### 4.4.3 $L_2$ -Norm Fit

Another method to find  $(\sigma, \phi)$  is to use the  $L_2$ -norm, whose advantage is the low computational complexity. The optimal parameter in the  $L_2$  sense is computed using a simple line search on the inlier scale parameter  $\sigma$ . Unfortunately, the  $L_2$ -norm approach does not guarantee that the inlier proportion  $\phi$  is contained in the  $[0; 1]$  interval, and is not, in general, a “natural” measurement for distributions.

The optimal parameters in the  $L_2$  sense are found by solving the following equation:

$$(\sigma, \phi) = \arg \min \|\mathbf{H}_e - \mathbf{H}_m\|_2$$

under the constraint that:

$$\phi \in [0; 1].$$

If the inlier scale parameter  $\sigma$  is known, the inlier proportion  $\phi$  can be found by taking the minimum mean square solution of the following equation system:

$$\mathbf{H}_e = \phi \cdot \mathbf{H}_I + (1 - \phi) \cdot \mathbf{H}_O,$$

i.e.,

$$\phi = \frac{(\mathbf{H}_I - \mathbf{H}_O)^T (\mathbf{H}_e - \mathbf{H}_O)}{\|\mathbf{H}_I - \mathbf{H}_O\|^2}.$$

In practice,  $\sigma$  is found using an exhaustive search given a maximum a priori value, which in our case is set to 20 percent of the image’s dynamic range.<sup>3</sup>

#### 4.4.4 Results

To illustrate the behavior of each fitting method, two pictures of a pedestrian street are shown. The outliers are the pedestrians along with their shades and the inliers are the background regions of the scene. The outlier proportion is varied by extracting from the image in Figs. 5a and 5b several subimages, which contain less and less background information, as illustrated in Figs. 5c, 5d, 5e, and 5f. Table 1 shows the result of fitting the OutlierMix model to the error histogram. The sensitive parameter is the inlier scale since the background is similar for every image pair, the inlier scale value should not change within the table.<sup>4</sup> The  $L_2$  estimator differs slightly from the maximum likelihood, but gave also very satisfactory results. Although the relative differences between the estimates of the inlier scale appears to be large, its absolute value is very small compared to the dynamic range of the error ( $[-1; 1]$ ).

The outlier modeling is also compared to the UniformMix model, which assumes that the outliers can be represented by a uniform distribution. The results shown in Table 2 are not as good as the ones using the OutlierMix modeling, but are still better than the traditional median-based approach.

The modeling fit is shown in Fig. 10 for the OutlierMix model using the maximum likelihood estimator. The fit is only shown for the maximum-likelihood estimator because, graphically, there is hardly a difference with the  $L_2$  estimator. Fig. 11 shows the *outlier mask*. The outlier mask represents, for each pixel, the probability to belong to the outliers—defined by (14)—where the gray scale is linearly related to the probability to be an outlier (black is used for high probability of outliers). We can see that the mask is able to discriminate the inliers from the outliers even in the presence of more than 95 percent outliers.

#### 4.5 Motion Model Based on the OutlierMix model

From the OutlierMix model, we can derive a new  $M$ -estimator [5] by associating a weight  $W(r)$  to the error value  $r$  of each pixel, according to whether the pixel is an outlier or not. Specifically, we set  $W(r)$  equal to its probability to belong to the inliers:

$$\begin{aligned} W(r) &= P(\text{inlier} | r) = \frac{P(\text{inlier}) \cdot P(r | \text{inlier})}{P(r)} \\ &= \frac{\phi H_I(\sigma, r)}{\phi H_I(\sigma, r) + (1 - \phi) H_O(r)}. \quad (19) \end{aligned}$$

In the context of motion estimation, the weight factor is introduced by the derivative  $\frac{\partial l(r_i)}{\partial r_i}$  of the objective function defined in Section 4.1. We set this derivative equal to the weight of (19):

3. Special care has to be taken in finding  $\sigma$  to avoid possible local minima.

4. Strictly speaking, some little changes could occur since it is not exactly the same background due to the different framing.



Fig. 5. (a) and (b) Original pictures used to perform outlier detection tests. (c), (d), (e), and (f) Ghost pictures used to perform the outlier tests. The images are superimposed using transparency; hence, the outliers appear as a ghost image. (c) The whole image, (d) region of picture (c), (e) region of picture (d), and (f) region of picture (e). The subimages are chosen such that the part of the image covered by the outliers is increasing.

$$\frac{\dot{\rho}(r_i)}{r_i} = W(r),$$

and compute an objective function from it:

$$\begin{aligned} \rho(r) &= \int_0^r \dot{\rho}(e) de \\ &= \int_0^r \frac{\dot{\rho}(e)}{e} \cdot e \cdot de \\ &= \int_0^r W(e) \cdot e \cdot de. \end{aligned} \quad (20)$$

The objective function is set to 0 at the origin:  $\rho(0) = 0$  and is computed numerically. The motion estimator using the OutlierMix model minimizes the objective function in (20). Note that this function is often asymmetric.

#### 4.6 Robustness of the OutlierMix Model

The traditional robustness computation approach evaluates the robustness of an estimator provided that a certain amount of samples are replaced by arbitrary values [3]. Unfortunately, it is not possible to characterize the performance of the OutlierMix model using this approach, because our model knows a priori all the possible values that the error can take.

TABLE 1  
Inlier Scale ( $\sigma$ ) and Outlier Percentage ( $1 - \phi$ ) Estimation

Method	Median	$L_2$ -norm		Max-likelihood	
Parameter	$\sigma$	$1 - \phi$	$\sigma$	$1 - \phi$	$\sigma$
Whole image (c)	0.028	22%	0.024	27%	0.020
sub-image (d)	0.039	37%	0.024	42%	0.019
sub-image (e)	0.061	52%	0.023	56%	0.018
sub-image (f)	1.192	97%	0.021	98%	0.014

The table compares three methods that use the OutlierMix model to determine the inlier scale and the outlier percentage of the images in Fig. 5. The inlier scale, which should be approximately constant, is also compared to a standard method that uses the median estimator. The two methods deliver similar results and are superior to the median-based estimator.

Consequently, it does not fit in the traditional framework of robustness characterization. Nevertheless, in our context, we can argue that the model can handle an arbitrary percentage of outlying data, since the outlier detection is just a matter of choosing the proportion among two distributions. Hence, we propose two experiments to measure the performance of the OutlierMix model.

The first experiment evaluates the robustness of the scale estimate ( $\sigma$ ) to outliers. Table 1 shows the estimate of the error scale for the pictures of Fig. 5. The error scale should remain approximately constant. We can see that the estimate derived from the error median—the method generally used by the standard estimators—is stable for the cases of less than 50 percent outliers. Above 50 percent, it becomes unstable and totally fails in presence of 98 percent of outliers. The same phenomenon is illustrated in Fig. 6, where the weighting factor of the Lorentzian model, defined in (8) is shown for every pixel. The figure shows the result obtained considering Figs. 5e and 5f. The first column of Fig. 6 shows the weight based on the scale found with the OutlierMix model fit with the  $L_2$ -norm and the second column shows the weight based on the scale computed with the error median. More details are visible in the outlier of Fig. 6b than in Fig. 6a, thus the  $M$ -estimator gives a variable weighting to the pixels of the outlier, instead of rejecting them uniformly, as in Fig. 6a. More striking is the difference between Figs. 6c and 6d, where the traditional estimator is unable to distinguish the inliers from the outliers. This shows that the scale estimate using the OutlierMix model is more robust.

The second experiment compares the performance of a translation estimation with an  $M$ -estimator using a Lorentzian and Geman-McClur distribution (see Section 4.1) to

TABLE 2  
Inlier Scale ( $\sigma$ ) and Outlier Percentage ( $1 - \phi$ ) Estimation

Method	$L_2$ -norm		Max-likelihood	
Parameter	$1 - \phi$	$\sigma$	$1 - \phi$	$\sigma$
Whole image (c)	19%	0.025	20%	0.025
sub-image (d)	37%	0.024	34%	0.023
sub-image (e)	42%	0.026	45%	0.024
sub-image (f)	98%	0.013	98%	0.012

The table compares two methods that use the UniformMix model to determine the inlier scale and the outlier percentage of the images in Fig. 5. The last line shows that the inlier scale estimate, which should be approximately constant, is not as robust as in the experiment in Table 1.

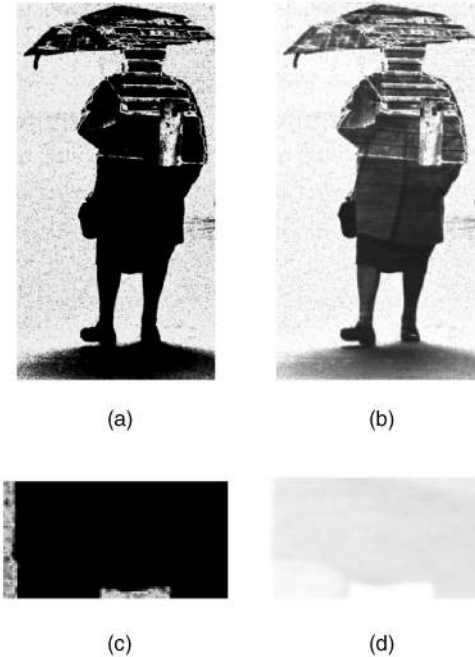


Fig. 6. Comparison of outlier masks for the OutlierMix model in (a) and (c) and a traditional  $M$ -estimator using the median to compute the error scale in (b) and (d). We can see in (d) that the traditional estimator fails to distinguish the outliers from the inliers. These masks are computed from the images in Fig. 5 and are rendered using a 2.2 gamma correction.

the motion estimator using the OutlierMix and the UniformMix models. The translation is defined as

$$\mathbf{p}' = \mathbf{p} + \theta,$$

where  $\mathbf{p}$  and  $\mathbf{p}'$  are the matching positions in the image pair, and  $\theta$  is the unknown (two dimensional) translation parameter. Two pictures of the same scene are taken in the same conditions with a fixed camera. One of the pictures has been corrupted by taking the left part of the picture and copying it to the right as illustrated in Fig. 8b. By extracting a subimage pair from these images, we can control the amount of outliers present in the pair. Then, for a whole set of image pairs, a motion estimation is performed,<sup>5</sup> starting from a misalignment of 15 pixels in amplitude. The results—shown in Fig. 7a—are binary: either the algorithm converges and the error is insignificant or it does not converge and the resulting error is large. The misregistration is measured as the distance between the real position of the image (known a priori) and the one delivered by the pose estimation. Five tests are conducted<sup>6</sup> and the graph shows the median of the results. The model using the OutlierMix model clearly outperforms the traditional ones, and is able to handle 86 percent to 90 percent outliers in this particular example.<sup>7</sup> For the same example, the Gaussian  $M$ -estimator handles up to 32 percent outliers, the Lorentzian  $M$ -estimator handles up to 60 percent outliers, and the Geman-McClur  $M$ -estimator handles up to 64 percent outliers. The UniformMix model handles up to 86 percent outliers and is therefore better than the traditional

5. The estimation is performed with a Gaussian multiresolution pyramid with four levels.

6. Five different initial conditions are used, each of them with the same misalignment amplitude.

7. The randomness of the experiment made the estimator fail at 88 percent four times out of five, but succeeded at 90 percent three times out of five.



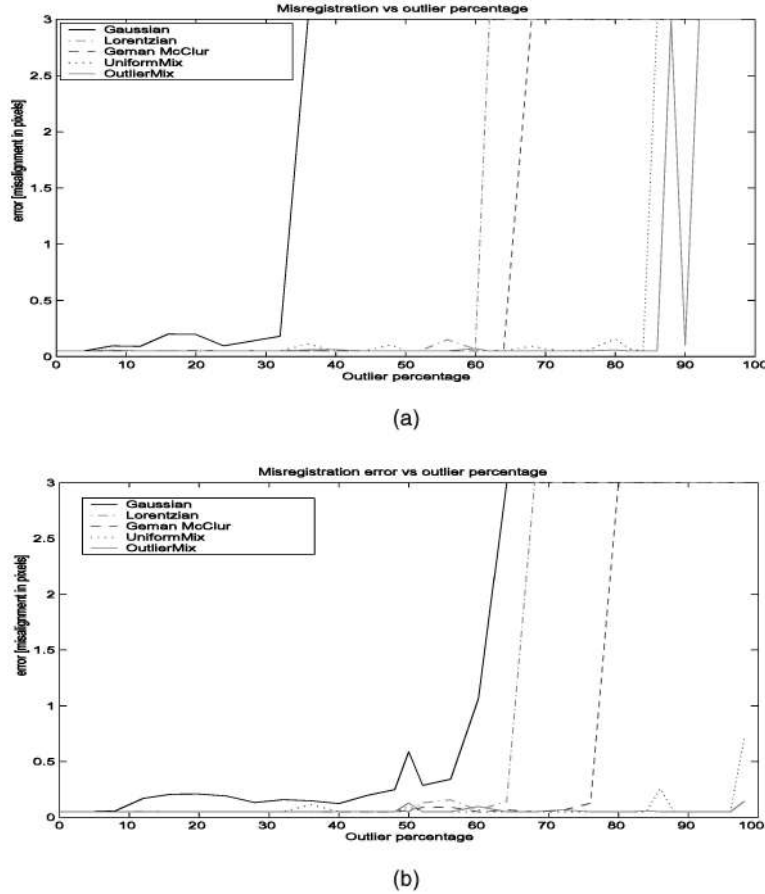


Fig. 7. Comparison of different estimators measured for two initial conditions. In (a), 15 pixels of initial misalignment—median result for five tests. In (b), 1.5 pixels of initial misalignment. A translation is estimated between two pictures, and the final translation error is shown. The OutlierMix model is the most robust of the set, handling up to 90 percent outliers in the first case and 98 percent in the second. There is some randomness associated with the test: for example the outlier-based model is able to handle 90 percent of outliers but fails with 88 percent.

$M$ -estimators but worse than the OutlierMix model. We want to emphasize that this model is sometimes used without estimating the outlier proportion  $(1 - \phi)$  [12], which is one of the key elements in the success of this implementation. The experiment is reproduced for an initial misalignment of 1.5 pixels in amplitude, showing similar results.

In general, the performance of the OutlierMix model depends on the image content: the more the outlier data differs from the inlier data, the better the OutlierMix model performs compared to a traditional approach. Indeed, having a large discrepancy between the inliers and outliers

facilitates the inliers/outliers mixture problem, but impairs the median estimator used in traditional techniques. We found, experimentally, that in the worst case, the outlierMix model behaves like a heavy tailed  $M$ -estimator as the Geman-McClur reviewed in this paper.

#### 4.7 Computational Complexity and Estimation Efficiency

The OutlierMix model is approximately as complex, in terms of computational cost, as the usual  $M$ -estimators. In addition to the computation required by the  $M$ -estimators,



Fig. 8. Images used to measure the robustness to outliers of different estimators shown in Fig. 7. The right part of (b) is corrupted and the outlier percentage is chosen by extracting a subimage pair and carefully controlling the framing of the extraction.

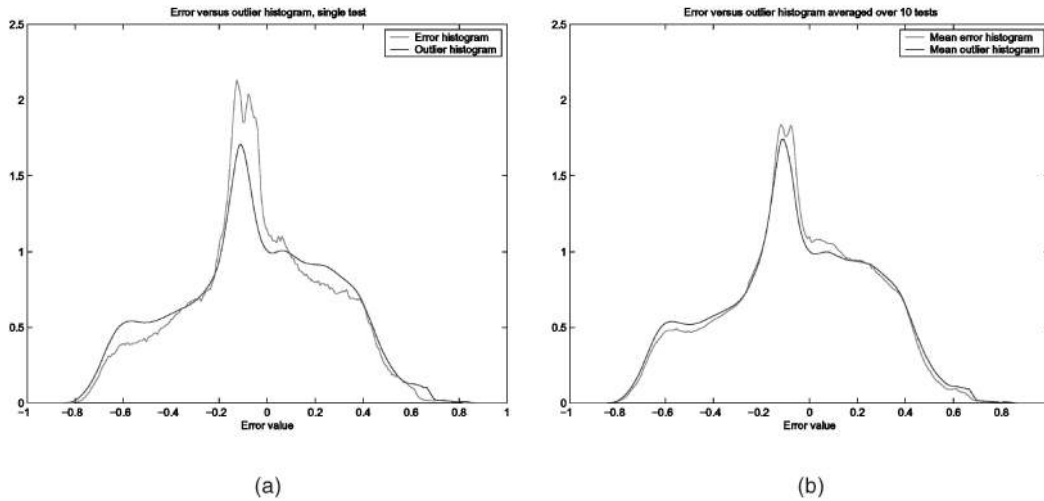


Fig. 9. Comparison between the predicted outlier histogram—smooth (blue) curve—and the measured error distribution of the superposition of the pictures depicted in Fig. 2d. The whole figure is used to compute the histograms. (a) In some locations, the data exhibits peak values that are not present in the modeling. These peak values are generated by the uniform areas contained in the images. Because of the uniformness of these areas, the matching does not, in general, lead to the expectation of the matching, thus explaining the divergence of the model from the data. (b) Shows the average over 10 experiments showing that the error histograms tend toward the outlier model.

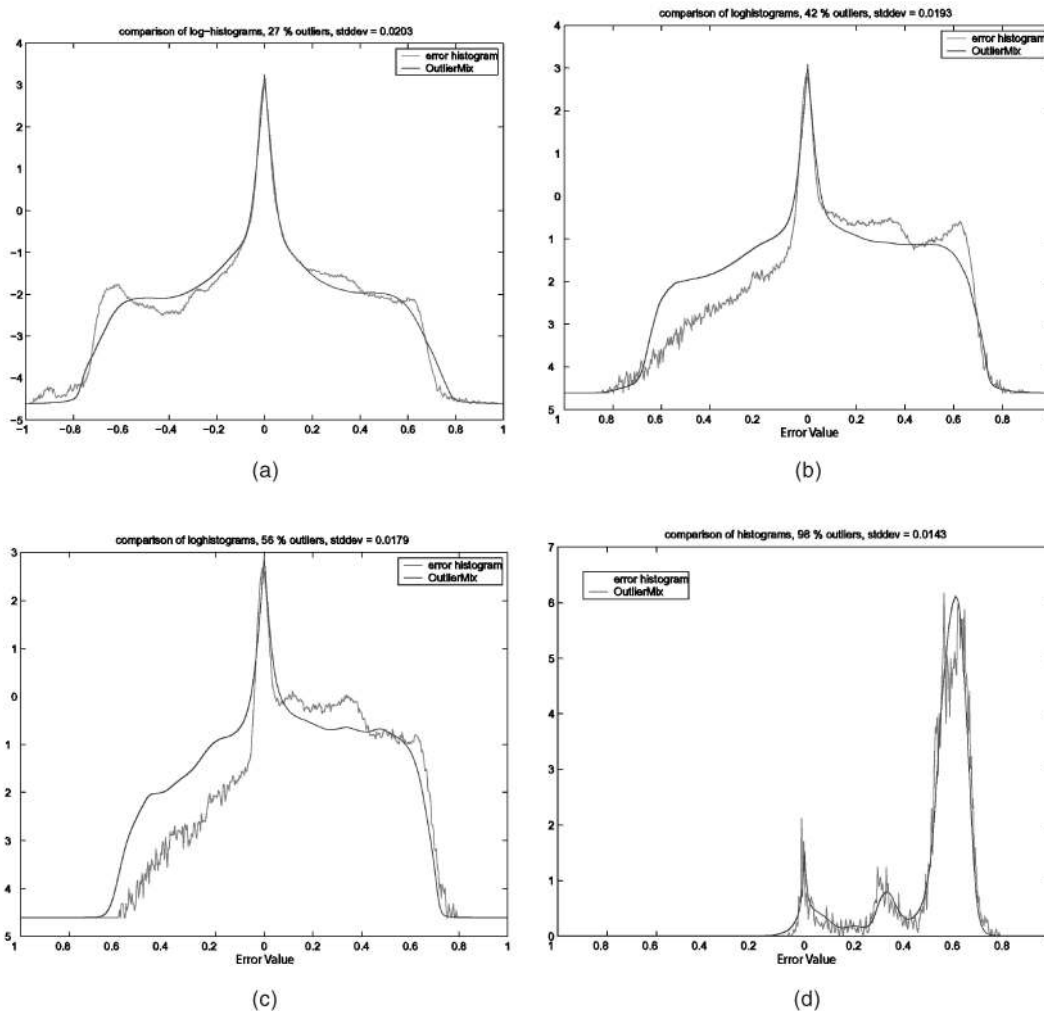


Fig. 10. Fitting the OutlierMix model to the error histogram with a maximum-likelihood approach. (a) Shows 27 percent outliers, (b) shows 42 percent outliers, (c) shows 56 percent outliers, and (d) shows 98 percent outliers. The graphs correspond to the images of Fig. 5 and are shown on a logarithmic vertical scale to graphically emphasize the error, except for (d). The smooth (blue) curve shows the OutlierMix model.

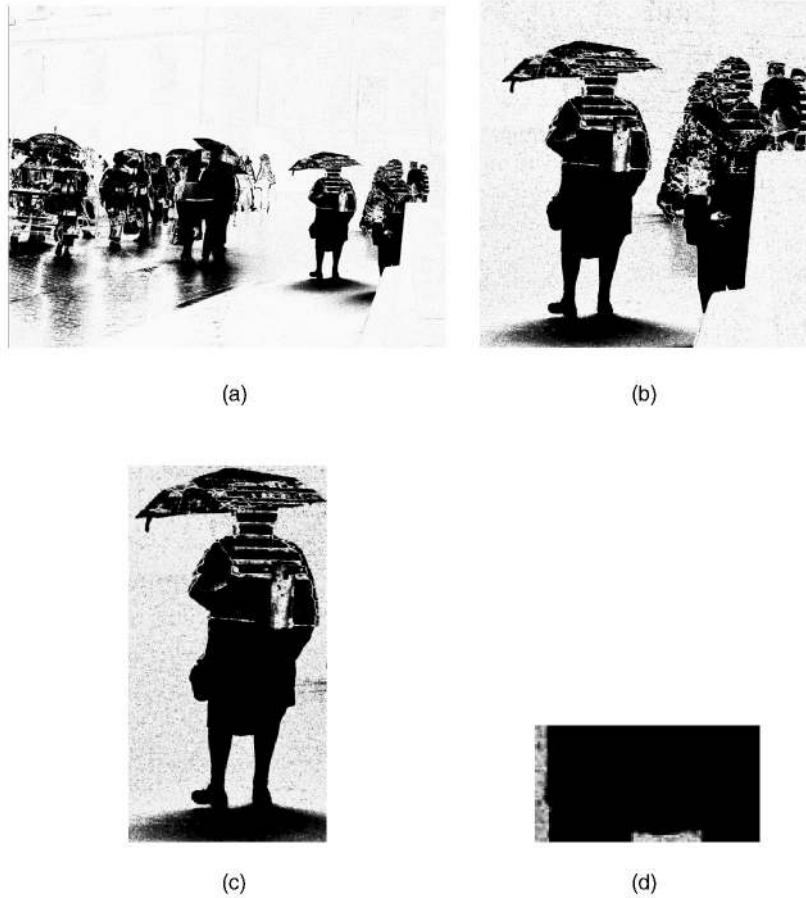


Fig. 11. Outlier masks associated to the images in Fig. 5. The gray scale is proportional to the probability to be an outlier; black indicates a high probability (a 2.2  $\gamma$  correction is used to render the images).

the OutlierMix model has to compute two image histograms (once), and requires an extensive search on the scale parameter of the inlier model. Nevertheless, the extensive search is based on histograms that contain in practice around 500 samples,<sup>8</sup> which is very small in comparison to the number of pixels in an image. Additionally, a better estimate of the scale parameter leads to faster convergence; hence, the algorithm needs fewer steps to converge to the solution. The UniformMix model has the same complexity than the OutlierMix model, if we do not take the two initial image histograms into account.

A nice property of the OutlierMix estimator is its relative efficiency. The relative efficiency is defined as the ratio between the lowest achievable variance for the estimated parameters (i.e., the Cramer-Rao lower Bound) and the actual variance provided by the given method [4]. In the absence of outliers, the outlier-based model reduces to a standard least-square estimation that reaches the Cramer-Rao lower bound. In other words, the OutlierMix motion model achieves a relative efficiency of 1 in absence of outliers. Indeed, the probability to be an inlier becomes 1 in this case, leading to a weight  $W(r) = 1$  in (20) that generates an objective function  $\rho(r) = r^2$ .

8. On an 8bits/sample image, the error values range from -255 to 255, making a total of 511 possible error values.

## 5 LIMITATIONS

In the examples shown so far, the outlier model of Section 3 has proven to explain the data quite well. There is, however, a particular case where the model does not fit the data very well, which occurs when the image contains a large uniform area. For example, in Fig. 2d, more than 1/3 of the image is covered by a uniform blue sky. To get the result of Figs. 2a, 2b, and 2c, we discounted the influence of the sky. If we take the sky into account, the error histogram exhibits a peak value that is not as accentuated in the outlier model. In this case, the expectation of the error distribution is not equal to the error distribution of a single realization of the comparison process. Indeed, when the areas of the pictures are too uniform, the error distribution tends to be sensitive to the way the pictures are compared. Nevertheless, except for the peak values, the model still fits the data quite well, as shown in Fig. 9. This also explains why the fit in Fig. 1 is better than the fit in Fig. 2. Now, if we conduct several different superpositions with the images of Fig. 2d and take the average of the resulting error histograms, then the error histogram tends to approach the outlier model, showing that the divergence observed in Fig. 9a is indeed due to an expectation problem. Fig. 9b shows the average of 10 error histograms obtained using 10 independent random superpositions.

Fig. 12 shows another example of an image superposition containing large uniform areas. The images are superimposed

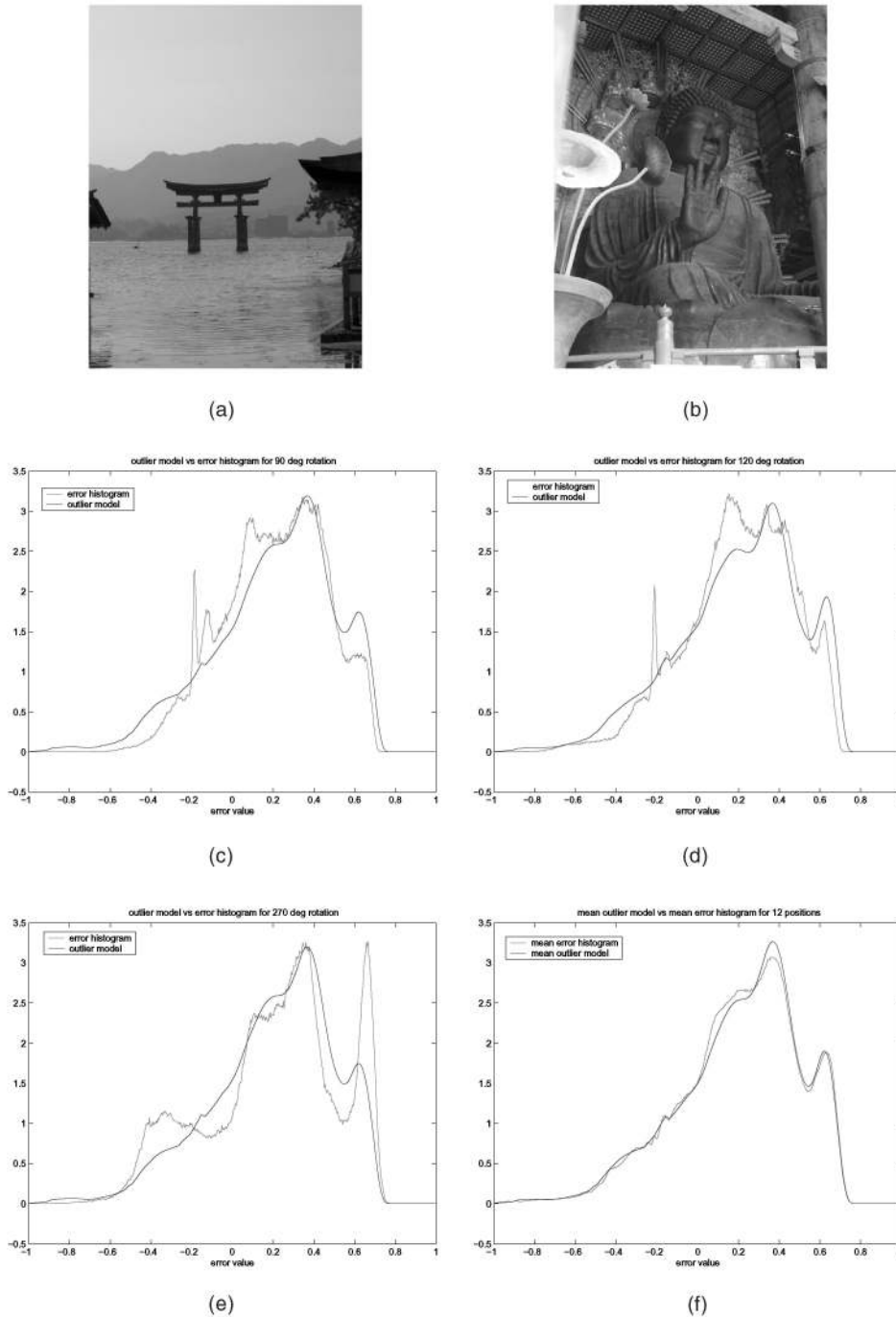


Fig. 12. Outlier model—smooth (blue) curve—versus error histogram: variations of the error histogram depending on the geometric configuration. (c), (d), and (e) are computed using three different angular positions of image. (b) over image (a). Because of the large uniform areas in (a), the error histogram varies substantially. (f) Shows the average over 12 tests.

by rotating one of the images. Fig. 12 shows the result for three different angular positions and for the average of the results. Note that, for each superposition, the outlier model changes slightly, because the histograms are computed on varying overlap areas. We want to emphasize that the Fig. 12 shows results whose performance are below average.

## 6 CONCLUSIONS

In this paper, we present a new way of calculating outliers in image pairs. The method differs from the traditional

approach by characterizing the outliers with a distribution computed from the initial images. This restriction in the outlier characterization allows the description of a new motion estimator that treats the pose estimation as a mixture of inliers versus outliers and is able to handle outlier percentages that exceed 50 percent of the image. The model has been tested using two kinds of experiments: The first experiment tests the ability of the model to discriminate the outliers from the inliers using two pictures taken with a fixed camera. The second experiment tests the performance of the motion estimator derived from the

OutlierMix model. Both experiments show a substantial improvement compared to the standard techniques in use.

These results can also serve different purposes: For segmentation applications, or coding applications, the OutlierMix model can be used to separate the moving objects from the background without using arbitrary thresholds on the error values.

Further research will investigate the extension of this modeling to color images. An interesting extension would also be to consider multiresolution to account for local relationships of pixel values.

## APPENDIX A

### OUTLIER MODEL COMPUTATIONS

This appendix details the assumptions involved in the OutlierMix model. As a convention we use capital letters for random variables ( $\Theta_0$ ) and lowercase letters for the realizations of these random variables ( $\theta_0$ ).

Let  $\Theta_0$  and  $\Theta_1$  be two vectors of independent discrete random variables. Let  $I_0$  and  $I_1$  be two injective functions ( $I_0$  and  $I_1$  are the images that are compared).

We are interested in the probability of error  $P(r)$  when comparing two images:

$$P(r) = \Pr\{I_0(\theta_0, \mathbf{p}) - I_1(\theta_1, \mathbf{p}) = r\},$$

where  $\mathbf{p}$  is the position in the image ( $\mathbf{p}$  is not random), and  $r$  the error (or *residual*). By assuming that  $P(r)$  does not depend on the position  $\mathbf{p}$  in the image, the probability of error  $P(r)$  is equal (by definition) to the expectation  $E(\cdot)$  of the indicator function:

$$P(r) \triangleq E_{\Theta_0, \Theta_1} [\mathbf{1}_{\{I_0(\theta_0, \mathbf{p}) - I_1(\theta_1, \mathbf{p}) = r\}}] \forall \mathbf{p}, \quad (21)$$

where  $\mathbf{1}_{\{b\}}$  is the indicator function ( $\mathbf{1}_{\{b\}}$  is equal to 1 if  $b$  is true, 0 otherwise). The histogram  $H$  can be written as

$$H(r, \theta_0, \theta_1) \triangleq \frac{1}{M} \sum_{\mathbf{p}} \mathbf{1}_{\{I_0(\theta_0, \mathbf{p}) - I_1(\theta_1, \mathbf{p}) = r\}},$$

where  $M$  is the number of pixels. The expectation of the error histogram is equal to

$$E_{\Theta_0, \Theta_1} [H(r, \Theta_0, \Theta_1)] = E_{\Theta_0, \Theta_1} \left[ \frac{1}{M} \sum_{\mathbf{p}} \mathbf{1}_{\{I_0(\theta_0, \mathbf{p}) - I_1(\theta_1, \mathbf{p}) = r\}} \right].$$

Note that from (21), the expectation of the error histogram is equal to  $P(r)$ . Since the expectation is a linear operator and the sum is finite, we can switch the expectation and the sum:

$$E_{\Theta_0, \Theta_1} [H(r, \Theta_0, \Theta_1)] = \frac{1}{M} \sum_{\mathbf{p}} E_{\Theta_0, \Theta_1} [\mathbf{1}_{\{I_0(\theta_0, \mathbf{p}) - I_1(\theta_1, \mathbf{p}) = r\}}].$$

The expectation can be expressed as

$$E_{\Theta_0, \Theta_1} [\mathbf{1}_{\{I_0(\theta_0, \mathbf{p}) - I_1(\theta_1, \mathbf{p}) = r\}}] = \sum_{\theta_0} \sum_{\theta_1} \mathbf{1}_{\{I_0(\theta_0, \mathbf{p}) - I_1(\theta_1, \mathbf{p}) = r\}} P(\theta_0, \theta_1),$$

but the indicator function of the error can be expressed as a cross-correlation,

$$\mathbf{1}_{\{I_0(\theta_0, \mathbf{p}) - I_1(\theta_1, \mathbf{p}) = r\}} = \sum_{\forall u} \mathbf{1}_{\{I_0(\theta_0, \mathbf{p}) = u\}} \mathbf{1}_{\{I_1(\theta_1, \mathbf{p}) = u - r\}},$$

hence, the expectation is expressed as

$$\begin{aligned} & E_{\Theta_0, \Theta_1} [\mathbf{1}_{\{I_0(\theta_0, \mathbf{p}) - I_1(\theta_1, \mathbf{p}) = r\}}] \\ &= \sum_{\forall \theta_0} \sum_{\forall \theta_1} P(\theta_0, \theta_1) \sum_{\forall u} \mathbf{1}_{\{I_0(\theta_0, \mathbf{p}) = u\}} \mathbf{1}_{\{I_1(\theta_1, \mathbf{p}) = u - r\}} \\ &= \sum_{\forall u} \sum_{\forall \theta_0} \sum_{\forall \theta_1} P(\theta_0, \theta_1) \mathbf{1}_{\{I_0(\theta_0, \mathbf{p}) = u\}} \mathbf{1}_{\{I_1(\theta_1, \mathbf{p}) = u - r\}}, \end{aligned}$$

by assuming that every sum is finite.

Because  $\Theta_0$  and  $\Theta_1$  are independent, we can separate the probabilities of  $\Theta_0$  and  $\Theta_1$ :

$$P(\theta_0, \theta_1) = P(\theta_0)P(\theta_1),$$

thus, the expectation can be expressed as a cross-correlation:

$$\begin{aligned} & E_{\Theta_0, \Theta_1} [\mathbf{1}_{\{I_0(\theta_0, \mathbf{p}) - I_1(\theta_1, \mathbf{p}) = r\}}] \\ &= \sum_{\forall u} \left[ \sum_{\forall \theta_0} P(\theta_0) \mathbf{1}_{\{I_0(\theta_0, \mathbf{p}) = u\}} \sum_{\forall \theta_1} P(\theta_1) \mathbf{1}_{\{I_1(\theta_1, \mathbf{p}) = u - r\}} \right], \\ &= \sum_{\forall u} [\Pr\{I_0(\theta_0, \mathbf{p}) = u\} \cdot \Pr\{I_1(\theta_1, \mathbf{p}) = r - u\}]. \end{aligned}$$

Finally,

$$\begin{aligned} & E_{\Theta_0, \Theta_1} [H(r, \Theta_0, \Theta_1)] = \\ & \frac{1}{M} \sum_{\mathbf{p}} \sum_{\forall u} [\Pr\{I_0(\theta_0, \mathbf{p}) = u\} \cdot \Pr\{I_1(\theta_1, \mathbf{p}) = r - u\}]. \end{aligned}$$

Now, by assuming that the statistics of the pixel in an image does not depend on the position ( $\mathbf{p}$ ) where the pixel is located in the image, the expectation of the error histogram is equal to the cross-correlation of the pixel value distribution:

$$\begin{aligned} & E_{\Theta_0, \Theta_1} [H(r, \Theta_0, \Theta_1)] = \\ & \sum_{\forall u} [\Pr\{I_0(\theta_0) = u\} \cdot \Pr\{I_1(\theta_1) = r - u\}]. \end{aligned}$$

By further assuming that the probability density function of the pixel is given by the image histogram

$$\frac{1}{M} \sum_{\mathbf{p}} \mathbf{1}_{\{I_0(\theta_0, \mathbf{p}) = u\}} = \Pr\{I_0(\theta_0) = u\},$$

the expectation of the error histogram ( $H$ ) is equal to the cross-correlation of the histograms ( $H_0, H_1$ ) of the images:

$$E_{\Theta_0, \Theta_1} [H(r, \Theta_0, \Theta_1)] = \sum_{\forall u} H_0(u) \cdot H_1(r - u).$$

Finally, by assuming that the comparison process is constant (or that the error histogram does not depend on the image positioning), we get

$$E_{\Theta_0, \Theta_1} [H(r, \Theta_0, \Theta_1)] = H(r, \theta_0, \theta_1), \quad \forall \theta_0, \theta_1, \quad (22)$$

which finally leads to

$$H(r, \theta_0, \theta_1) = \sum_{\forall u} H_0(u) \cdot H_1(r - u).$$

The error histogram is equal to the cross-correlation of the image histograms.

## ACKNOWLEDGMENTS

The authors would like to thank Andrea Ridolfi for his help with Appendix A and the anonymous reviewers whose comments were very helpful in preparing a clearer presentation of their research.

## REFERENCES

- [1] B.G. Lindsay, "Mixture Models: Theory, Geometry and Applications," vol. 5, *NSF-CBMS Regional Conf. Series in Probability and Statistics*, 1995
- [2] G. McLachlan and D. Peel, *Finite Mixture Models*. New York: John Wiley and Sons, 2000.
- [3] P. Huber, *Robust Statistics*. New York: Wiley-Interscience, 1981.
- [4] S. Ayer, "Sequential and Competitive Methods for Estimation of Multiple Motions," PhD thesis, Swiss Fed. Inst. of Technology (EPFL), 1995.
- [5] P.J. Rousseeuw and A.M. Leroy, "Robust Regression and Outlier Detection," *Applied Probability and Statistics*, New York: John Wiley and Sons, Inc. 1987.
- [6] H.S. Sawhney and S. Ayer, "Compact Representations of Videos through Dominant and Multiple Motion Estimation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, no. 8, pp. 814-830, Aug. 1996.
- [7] G.D. Hager and P.N. Belhumeur, "Efficient Region Tracking with Parametric Models of Geometry and Illumination," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, no. 10, pp. 1025-1029, Oct. 1998.
- [8] J.-I. Park, N. Yagi, K. Enami, K. Aizawa, and M. Hatori, "Estimation of Camera Parameters from Image Sequence for Model-Based Video Coding," *IEEE Trans. Circuit and Systems for Video Technology*, vol. 4, no. 3, pp. 288-296, 1994.
- [9] V.L. Brailovsky, "An Approach to Outlier Detection Based on Bayesian Probabilistic Model," *Proc. Int'l Conf. Pattern Recognition*, pp. 70-74, 1996.
- [10] N.S. Netanyahu and I. Weiss, "Analytic Outlier Removal in Line Fitting," *Proc. 12th IAPR Int'l Conf. Computer Vision and Image Processing*, vol. 2B, pp. 406-408, 1994.
- [11] P. Schroeter, J.-M. Vesin, T. Langenberger, and R. Meuli, "Robust Parameter Estimation of Intensity Distributions for Brain Magnetic Resonance Images," *IEEE Trans. Medical Imaging*, vol. 17, no. 2, pp. 172-186, 1998.
- [12] P.S. Torr, R. Szeliski, and P. Anandan, "An Integrated Bayesian Approach to Layer Extraction from Image Sequences," *Proc. Int'l Conf. Computer Vision*, 1999.
- [13] J.M. Mendel, *Lessons in Estimation Theory for Signal Processing, Communications, and Control*, second ed. New York: Prentice Hall, 1995.
- [14] C.F. Olson, "Maximum-Likelihood Template Matching," *Proc. Computer Vision and Pattern Recognition*, 2000.
- [15] X. Gao and T. Boult, "Statistics of Natural Images and Models," *Proc. Computer Vision and Pattern Recognition*, 2000.
- [16] D. Hasler, "Perspectives on Panoramic Photography," PhD thesis, Swiss Fed. Inst. of Technology (EPFL), 2001.
- [17] A. Leon-Garcia, *Probability and Random Processes for Electrical Engineering*, second ed. Addison-Wesley 1994.
- [18] B. Aiazzi, L. Alparone, and S. Baronti, "Estimation Based on Entropy Matching for Generalized Gaussian pdf Modeling," *IEEE Signal Processing Letters*, vol. 6, no. 6, pp. 138-140, 1999.



**David Hasler** received the MS degree in micro-engineering, in 1996, and the PhD in communication systems in 2001, both from the Swiss Federal Institute of Technology in Lausanne (EPFL). During his undergraduate studies, he was an exchange student at the Carnegie Mellon University in Pittsburgh, Pennsylvania, and did his master's thesis at Daimler-Chrysler research, in Ulm, Germany, working on sensor simulation. He then joined the graduate school in Communication Systems at EPFL where he was awarded with a three year PhD fellowship. During his studies, he also worked at NASA Ames, California, on a mobile robot field test, and at Sony Research, California, on high-dynamic range imaging. In early 2002, he joined a startup company—Genimedia SA—and worked on perceptual image quality metrics. His main research area is computer vision and digital photography, with emphasis on color. He is a member of the IEEE.



**Luciano Sbaiz** received the "Laurea in Ingegneria Elettronica" degree from the University of Padova in 1993 (score 110/110 with honors). The object of his graduation thesis was the implementation and testing of an image coder based on the wavelet transform and using an efficient prediction technique of the wavelet coefficients. In 1994, he obtained a fellowship from Seleo S.p.A., Pordenone, Italy. From 1994 to 1997, he has worked on the Doctorate degree in electronics and telecommunications in the Dipartimento di Elettronica e Informatica, University of Padova, with advisor Professor Gian Antonio Mian. The topic of his research concerned the application of computer vision motion estimation algorithms to object oriented and model based video coders. The algorithm that has been developed is based on a representation of the scene by means of objects that undergo rigid motion. The motion equations are arranged in the form of a dynamic system whose state, which contains the motion parameters, is estimated by means of an extended implicit Kalman filter. The parameterization of the model allows for a very efficient coding of motion information in comparison to more common block-based schemes. Moreover, the structure imposed by the model implies that the reconstructed motion is very natural. During the years from 1997 to 1999, he worked as assistant at the Laboratory of Audio Visual Communications (LCAV) of the Swiss Federal Institute of Technology (EPFL) of Lausanne. His research concerned the development of computer vision techniques with application to the creation of video special effects. Since September 1999, he works with Dartfish SA, Fribourg, Switzerland. His tasks concern the development of algorithms for video processing and motion estimation.



**Sabine Süsstrunk** received the BS degree in scientific photography from the Swiss Federal Institute of Technology (ETH) and the MS degree in graphic arts publishing from the Rochester Institute of Technology (RIT). She is currently pursuing a PhD part-time at the School of Information System, University of East Anglia, Norwich, United Kingdom. Currently, she is an assistant professor for the Imaging and Visual Representation in the Audiovisual Communica-

tions Laboratory, Faculty of Information and Communication Sciences, at the Swiss Federal Institute of Technology (EPFL) in Lausanne, Switzerland. Her main research areas are color imaging, image quality metrics, and digital archiving. She is an active member of ISO/TC42/WG18 and JWG20-electronic imaging standards groups, and a vice president of the Society of Imaging Science and Technology (IS&T).



**Martin Vetterli** received the Dipl. El.-Ing. degree from ETH Zurich (ETHZ), Switzerland, in 1981, the MS degree from Stanford University in 1982, and the Doctorat ès Sciences degree from EPF Lausanne (EPFL), Switzerland, in 1986. He was a research assistant at Stanford and EPFL, and has worked for Siemens and AT&T Bell Laboratories. In 1986, he joined Columbia University in New York, where he was last an associate professor of electrical engineering and codirector of the Image and Advanced Television Laboratory. In 1993, he joined the University of California at Berkeley, where he was a professor in the Department of Electrical Engineering and Computer Sciences until 1997, and now holds an adjunct professor position. Since 1995, he has been a professor of communication systems at EPF Lausanne, Switzerland, where he chaired the Communications Systems Division (1996 and 1997), and heads the Audiovisual Communications Laboratory. Since 2001, he has directed the National Competence Center in Research on mobile information and communication systems. He has held visiting positions at ETHZ (1990) and Stanford (1998). He is a fellow of the IEEE, a member of SIAM, and was the area editor for speech, image, video, and signal processing of the *IEEE Transactions on Communications*. He is also on the editorial boards of *Annals of Telecommunications*, *Applied and Computational Harmonic Analysis*, and *The Journal of Fourier Analysis and Application*. He received the Best Paper Award of EURASIP in 1984 for his paper on multidimensional subband coding, the Research Prize of the Brown Boveri Corporation (Switzerland) in 1986 for his doctoral thesis, the IEEE Signal Processing Society's Senior Awards in 1991 and in 1996 (for papers with D. LeGall and K. Ramchandran, respectively). He won the Swiss National Latsis Prize in 1996, the SPIE Presidential award in 1999, and the IEEE Signal Processing Technical Achievement Award in 2001. He is a member of the Swiss Council on Science and Technology. He was a plenary speaker at various conferences (e.g., 1992 IEEE International Conference on Acoustics, Speech and Signal Processing) and is the coauthor, with J. Kovacevic, of the book *Wavelets and Subband Coding*, (Prentice-Hall, 1995). He has published more than 85 journal papers on a variety of topics in signal/image processing and communications and holds seven patents. His research interests include sampling, wavelets, multirate signal processing, computational complexity, signal processing for communications, digital video processing, and joint source/channel coding.

▷ **For more information on this or any other computing topic, please visit our Digital Library at <http://computer.org/publications/dlib>.**