

Fast Algorithm for Robust Template Matching with M-estimators

Jiun-Hung Chen, Chu-Song Chen, and Yong-Sheng Chen

Institute of Information Science, Academia Sinica, Taipei, Taiwan

e-mail: song@iis.sinica.edu.tw

Abstract

In this paper, we propose a fast algorithm for speeding up the process of template matching that uses M-estimators for dealing with outliers. We propose a particular image hierarchy called the p -pyramid that can be exploited to generate a list of ascending lower bounds of the minimal matching errors when a non-decreasing robust error measure is adopted. Then, the set of lower bounds can be used to prune the search of the p -pyramid, and a fast algorithm is thereby developed in this paper. This fast algorithm ensures finding the global minimum of the robust template matching problem in which a non-decreasing M-estimator serves as an error measure. Experimental results demonstrate the effectiveness of our method.

Index Terms —Template matching, robust template matching, M-estimator, fast algorithm.

I. Introduction

Finding a pattern or template in a signal is an important problem for signal and image processing. This so-called *template matching* can be applied to many applications such as image and video coding, pattern recognition, and visual tracking. It is usually assumed in template matching that the signal segments of interests do not change their appearances very much. Hence, template matching based on the criteria such as the *sum of absolute difference* (SAD) or the *sum of squared difference* (SSD) is commonly adopted. The popularity of using template matching for applications of signal or image processing is mainly due to its ease of implementation together with the many fast algorithms that can be used to speed up the matching process for various applications [1,7-8,13,15,22,25-26,28-29,35,39].

In a cluttered environment, however, some outliers such as impulse noises or partial occlusions may occur during the matching processes. In this situation, the SAD and SSD criteria are no longer suitable for template matching because they treat the outliers and inliers evenly when calculating the

error measures. One possible remedy for this weakness is to use a robust criterion instead of SAD or SSD. For this, the M-estimator technique [2][16][30][38] is one of the most popular methods to solve the problem of robust parameter estimation, and has been applied in many studies [3-5,11-12,14,23,31-33,36]. The basic idea of the M-estimator technique is to limit the influence of outliers in the matching error. In principle, the effects of the outlier can be suppressed with the M-estimator technique and therefore better estimations are obtained.

A typical procedure for finding solutions with M-estimators is the iterative-reweight procedure [30]. In each iteration of this procedure, a weighted least-square problem is solved and then the weights are adjusted for the next iteration for further refinement. Hence, when applying the iterative-reweight procedure for robust template matching, in each iteration, another template matching problem must be solved based on a weighted SSD error measure, in addition to which, multiple iterations are also necessary. Therefore, the computation of robust error measures is very time-consuming, although more accurate results can be obtained by adopting a robust error measure instead of non-robust ones.

¹ In the past, many methods have been proposed to speed up the matching process where the simple SAD or SSD criterion is used. However, to our knowledge, no method has been addressed for speeding up the process of template matching where robust error measures are used. In this paper, we propose a fast method for solving this problem. We will present this method by assuming that a 2D signal (e.g., an image) is used. Nevertheless, our algorithm can be easily generalized for any d -dimensional signal, $d \in \mathbb{N}^+$.

On the other hand, there are already many methods for speeding up the process of template matching where non-robust error measures are used. These methods can be divided into two classes. The methods in the first class only find a local minimum while the ones in the second class definitely find the global minimum. In principle, almost all the methods in the first class formulate the template matching as a search problem and find a solution by adopting the greedy strategy. Examples include the three-step search algorithm [22], the gradient-descent based method [29] and others [1][8][13][15][28][35][39]. The genetic algorithm-based methods [8][28] or the simulated annealing-based method [35] may have chances of finding the global minimum if their parameters are set appropriately to the given problems, but can not ensure that it will always be found. In essence, since these methods do not guarantee finding the global minimum, they are generally faster than those ensuring the global optimality.

The methods in the second class guarantee finding the global minimum, and the main idea of this class is basically prune and search [26][25][7]. Hence, the main issue of this class of approaches is on how to design the search strategies for pruning unnecessary searching branches. The successive elimination algorithm proposed by Li and Salari [26] eliminates impossible sites successively during

the searching process by using lower bounds derived from the triangle inequality. Their method can guarantee obtaining the global minimum, as does the full search (FS) method, and it is more efficient. In [25], Lee and Chen extended this idea by using a block-sum-pyramid structure, where a set of ascending lower bounds can be derived and serve as useful guidelines to prune the search process. However, the performance of their method depends on the search order. Recently, Chen et al. further refined it by using a winner-update strategy [7], which is not only irrelevant to the search order but is also faster.

Similar to those methods that ensure finding the global minimum but using non-robust error measures introduced above [26][25][7], the method developed in this paper also adopts an inequality in a particular image hierarchy to speed up the template-matching process with robust error measures. In essence, a set of ascending lower bounds of the minimal matching error can be generated with our method as long as the robust error measure is non-decreasing. This set of lower bounds can then serve as useful guidelines for pruning the redundant branches of the searching process. In addition, our method can ensure finding the global minimum, as the FS method does for robust template matching.

This paper is organized as follows. Section II introduces the image hierarchy used in this work and the associated ascending lower bound list. Section III presents the search strategies and our main algorithm. Section IV shows some experimental results. Finally, some discussion and conclusions are given in Sections V and VI, respectively.

II. Problem Formulation, P -pyramid, and Fundamental Inequality

A. Problem Formulation

We denote $I(i,j)$ as the intensity at position (i,j) in an image I . Assume that I_1 and I_2 are $N \times N$ images. The sum of robust differences (SRD) between two images I_1 and I_2 is defined as follows:

$$SRD_{\rho,\sigma}(I_1, I_2) \equiv \sum_{0 \leq i, j \leq N-1} \rho(|I_1(i, j) - I_2(i, j)|, \sigma) \quad (1)$$

where $\rho(\cdot, \cdot) : \mathbb{R}^+ \cup \{0\} \times \mathbb{R}^+ \rightarrow \mathbb{R}^+ \cup \{0\}$ is a robust error measure (or a robust loss function) [2][16][38] and σ is a parameter controlling the shape of $\rho(\cdot, \cdot)$.² Typically, the robust error measure, $\rho(\cdot, \cdot)$, is selected according to how it reduces the influences of outliers. Given an image template I , whose size is $N \times N$ and an image F whose size is $(2W+N) \times (2W+N)$, the robust template matching

¹ Another problem of the iterative-reweight procedure is that it can not definitely find the global optimum.

² Some common robust error measures will be introduced in Section II. E. In particular, if $\rho(x, \sigma) = x^2$ (or x) for all σ , then (1) becomes SSD (or SAD).

problem is defined as finding the position (u^*, v^*) with the minimum SRD among all possible search positions in the image F .

$$\begin{aligned} (u^*, v^*) &\equiv \arg \min_{-W \leq u, v \leq W} \text{SRD}_{\rho, \sigma}(I_t, F_{u,v}) \\ &= \arg \min_{-W \leq u, v \leq W} \sum_{0 \leq i, j \leq N-1} \rho(|I_t(i, j) - F(i+u, j+v)|, \sigma) \end{aligned} \quad (2)$$

where $F_{u,v}$ is an $N \times N$ image block with its upper-left point being (u, v) in the image F and the number of search sites is $(2W+1) \times (2W+1)$.

B. P -feasible

To simplify the notation of the robust error measures, we define

$$\tau_\sigma(x) = \rho(x, \sigma) \text{ for all } x \in \mathbb{R}^+ \cup \{0\}, \quad (3)$$

and abbreviate $\tau_\sigma(x)$ to be $\tau(x)$ for the cases without ambiguity in the following.

Definition 1 [p -feasible]: A robust error measure $\tau(\cdot)$ is p -feasible if it satisfies both the following two conditions:

- (1) $\tau(\cdot)$ is non-decreasing, i.e., $a_1 \geq a_2$ implies $\tau(a_1) \geq \tau(a_2)$ for all $a_1, a_2 \in \mathbb{R}^+ \cup \{0\}$, and
- (2) for each pair of nonnegative values (a_1, a_2) , the following inequality with respect to the L_p -norm holds:

$$\forall a_1, a_2 \in \mathbb{R}^+ \cup \{0\}, \quad \tau(a_1) + \tau(a_2) \geq \tau(\|\mathbf{a}\|_p) \quad (4)$$

where $\mathbf{a} = [a_1 \ a_2]^T$ is a 2-D vector and $\|\mathbf{a}\|_p = (|a_1|^p + |a_2|^p)^{1/p}$ is the L_p -norm of \mathbf{a} , $p \in [1, \infty]$. In particular, $\|\mathbf{a}\|_\infty = \max(|a_1|, |a_2|)$.

Some properties associated with the p -feasible defined above are investigated below:

Property 1: Every non-decreasing $\tau(\cdot)$ is ∞ -feasible.

pf: This property can be easily derived according to the definitions.

Property 2: If a robust error measure $\tau(\cdot)$ is p -feasible, that implies it is also q -feasible for all $q \in [p, \infty]$.

pf: See Appendix A.

If a p -feasible robust error measure is used as the matching criterion, an ascending lower bound list of the matching errors can be obtained by constructing an image pyramid with respect to the L_p -norm, as described in the following.

C. p -pyramid and Fundamental Inequality

Assume that $N = 2^n$ ($n \in \mathbb{N}^+$). For each $N \times N$ image block $I = F_{u,v}$ ($-W \leq u, v \leq W$) that is contained in the image F , a p -pyramid of I is defined as a set of images $\{I^{0,p}, \dots, I^{m-1,p}, I^{m,p}, \dots, I^{n,p}\}$, where $I^{n,p} = I$ and the size of $I^{m,p}$ is $2^m \times 2^m$ ($0 \leq m \leq n$, $m \in \mathbb{N}^+ \cup \{0\}$). $I^{m,p}$ is referred to as the image on the level m of the p -pyramid. Level 0 and level n are called the highest and the lowest levels of the p -pyramid, respectively. Given an image $I^{m,p}$ on the level m , the image $I^{m-1,p}$ on the level $m-1$ is constructed using the following equation:

$$I^{m-1,p}(i, j) = \left\| \mathbf{I}_{i,j}^{m,p} \right\|_p \quad (5)$$

where

$$\mathbf{I}_{i,j}^{m,p} = \left[I^{m,p}(2i, 2j) \quad I^{m,p}(2i, 2j+1) \quad I^{m,p}(2i+1, 2j) \quad I^{m,p}(2i+1, 2j+1) \right]^T \text{ and } 0 \leq i, j \leq 2^{m-1} - 1.$$

Accordingly, the pyramids from level n to level 0 can be constructed iteratively. Totally, $(2W+1) \times (2W+1)$ p -pyramids are constructed from the image F . Figure 1 shows an illustration of the pyramids constructed from a 1-D signal.

Following the notions shown in (1), we define the SRD between I_1 and I_2 on the level m by $SRD_{\rho,\sigma}^{m,p}(I_1, I_2) = SRD_{\rho,\sigma}(I_1^{m,p}, I_2^{m,p})$ for $m = 0, 1, \dots, n$. Furthermore, we denote $SRD_{\tau}(I_1, I_2) = SRD_{\rho,\sigma}(I_1, I_2)$ if a fixed σ is considered (where $\tau(\cdot)$ is defined in (3)). Then, it can be shown that the following fundamental inequality holds for the image hierarchy defined above.

Theorem 1: Given a robust error measure $\tau(\cdot)$ that is p -feasible, then for all I_1, I_2 ,

$$SRD_{\tau}(I_1^{n,p}, I_2^{n,p}) \geq SRD_{\tau}(I_1^{n-1,p}, I_2^{n-1,p}) \geq \dots \geq SRD_{\tau}(I_1^{0,p}, I_2^{0,p}) \quad (6)$$

pf: See Appendix B.

Hence, given a image template I_t and an image block $F_{u,v}$ ($-W \leq u, v \leq W$), a set of ascending lower bounds, $SRD_{\tau}^{m,p}(F_{u,v}, I_t)$, $m=0, 1, \dots, n$ can be obtained according to Theorem 1.

Notice that the number of robust differences involved in the computation of each lower bound $SRD_{\tau}^{m,p}(F_{u,v}, I_t)$ ($0 \leq m < n$) is $2^m \times 2^m$, which is smaller than $2^n \times 2^n$, the number of robust differences required for $SRD_{\tau}^{n,p}(F_{u,v}, I_t)$. Therefore, the lower bounds can be computed more efficiently than the sum of robust differences with respect to the original image. In particular, the higher are the levels in a pyramid, the faster are the computations of the associated lower bounds. In fact, the ratio

of the required number of the computations of the robust differences on the level m to that required for the level n is $\frac{1}{4^{n-m}}$. Accordingly, even when all of the lower bounds are computed, the required time is less than $\sum_{m=1}^{n-1} \frac{1}{4^{n-m}} \leq \frac{1}{3}$ of that required for the level n , the original image block. Such an ascending lower bound list of the matching error can be used for speeding up the matching process by incorporating it into a systematic search strategy, as introduced in Section III.

Remark 1 [Efficient Construction of the P -pyramids of an Image]: A p -pyramid of each image block can be constructed independently by using (5). However, it is very time-consuming if each pyramid is constructed independently. In fact, the p -pyramids can be constructed more efficiently by considering the computation and storage redundancies between neighboring image blocks when constructing and storing the p -pyramids. We introduce this method by using an example, as shown in Figure 1. Considering node A in the first level image of Pyramid 1, as shown in Figure 1, one can observe that node A is also contained in Pyramid 3. Similarly, node B (or C) is shared by Pyramids 2 and 4 (or Pyramids 3 and 5). Hence, if each pyramid is constructed independently according to (5), the values of nodes A, B, and C will be computed twice, once for each pyramid. In our work, to remove this redundancy and to save computation and storage of the p -pyramids for all the image blocks, the method illustrated in Figure 1 is adopted. That is, I^1 , the image containing every node of the first-level images of all the pyramids, is first constructed based on I^2 . Similarly, I^0 can be constructed based on I^1 . After I^0 and I^1 are constructed, all the pyramids are then available, as shown in Figure 1. More details about efficient construction of pyramids can be found in [7].

Remark 2 [Free Sampling]: In the above description, a particular level of the p -pyramid is built with the L_p -norm of the 2×2 points from its lower neighbor level, as shown in (5). We refer to it as the 2-2 down-sampling in this case. In general, we can use m - n down-sampling instead of 2-2 down-sampling for all $m, n \in \mathbb{N}^+$, and the associated p -pyramid can be reconstructed in a similar way as well. Similarly, m down-sampling can be used for a 1-D signal and m - n - k down-sampling can be used for a 3-D signal, and so on, where $k \in \mathbb{N}^+$.

D. Characterization of P -feasible Robust Error Measures

In Section II.C, we have shown that if a p -feasible robust error measure is selected for template matching, then we can construct an ascending lower bound list associated with a particular image hierarchy, the p -pyramid. Before introducing the search procedure that exploits the series of lower bounds in detail, we illustrate an important issue about whether such a lower-bound list can be constructed. In particular, the following problem is worthy to be addressed:

“Given a robust error measure $\tau(\cdot)$, under what condition can it be ensured that we can always

find a $p \in [1, \infty]$ such that $\tau(\cdot)$ is p -feasible?”

In this paper, we tackle this problem in consideration of the class of non-decreasing robust error measures. In fact, Property 1 has shown that if a robust error measure is non-decreasing, it is ∞ -feasible. Therefore, when the ∞ -pyramids are built for both the template and the image blocks, an ascending lower bound list can then be constructed according to Theorem 1. Hence, it ensures that every non-decreasing robust error measure is p -feasible for some p because $p = \infty$ is a trivial solution.

³ In addition to $p = \infty$, let us further investigate the other p values that allow a robust error measure to be p -feasible. Given a non-decreasing robust error measure, $\tau(\cdot)$, let Γ_τ , the *feasible set* associated with τ , be defined as the set of values allowing $\tau(\cdot)$ to be p -feasible: $\Gamma_\tau = \{p \in [1, \infty] \mid \tau \text{ is } p\text{-feasible}\}$. By considering the maximal lower bound of Γ_τ , the following property can be derived:

Property 3: Given a $\tau(\cdot)$ that is non-decreasing, there exists a discriminative value p' such that $\tau(\cdot)$ is p -feasible for all $p \in (p', \infty]$ and is not p -feasible for all $p \in [1, p')$, where p' is the maximal lower bound of Γ_τ , the feasible set associated with τ .

pf: This property can be derived directly from Properties 1 and 2.

Hence, the feasible set associated with a robust error measure can be clearly specified with its discriminative value p' by further identifying the following two conditions: τ is p' -feasible or τ is not p' -feasible. If τ is p' -feasible, then $\Gamma_\tau = [p', \infty]$; otherwise, $\Gamma_\tau = (p', \infty]$. After specifying the feasible set, another problem worth consideration is

“Which p contained in the feasible set associated with a non-decreasing robust error measure is a better choice for speeding up the process of robust template matching?”

We investigate the above problem from the implementation point of view. In practice, to simplify the computation, it is better to select p as integers instead of floating-point numbers. When p is restricted to being an integer, the feasible set associated with $\tau(\cdot)$ can then be uniquely specified by $\Gamma_\tau = \{n \in \mathbb{N}^+ \cup \{\infty\} \mid n \geq n', n' \in \mathbb{N}^+ \cup \{\infty\}\}$, where $n' = \lceil p' \rceil$. In particular, we call $\tau(\cdot)$ to be *minimal n' -feasible* in this case, and n' is also referred to as the *minimal feasible value* of $\tau(\cdot)$. Note that the smaller is n , the less is the computational overhead of the L_n -norm for $n \in \mathbb{N}^+$. Therefore, a better choice of n is therefore $n = n'$. ⁴

³ How to construct a list of ascending lower bounds for the general class of robust error measures that are not necessary to be non-decreasing remains an open problem.

⁴ Although the computation of the ∞ -norm is also simple since only the absolute values and the $\max(\cdot, \cdot)$ operations are involved, we find that in practice its speedup performance is usually worse because the lower bounds associated with an ∞ -pyramid are usually not tight enough. An example is given in Section IV. A.

E. Minimal Feasible Value of Commonly Used Robust Error Measures

In the following, we will give a study of the minimal feasible values for some commonly used robust error measures. An interesting phenomenon shown below is that almost all commonly used non-decreasing robust error measures are minimal 1-feasible or minimal 2-feasible.

Given a robust error measure $\rho(\cdot, \cdot)$, we define $H_\rho = \{\rho(\cdot, \sigma) | \sigma \in \mathbb{R}^+\}$. The Huber's estimator $\rho_1(\cdot, \cdot)$ (see Figure 2(a)) has least squares behaviors for small residues, and the more robust least-absolute-values behavior for large residues [14][19][31][33]:

$$\rho_1(r, \sigma) = \begin{cases} \frac{r^2}{2} & \text{if } r \leq \sigma, \\ \sigma \left(r - \frac{\sigma}{2} \right) & \text{otherwise.} \end{cases} \quad (7)$$

Property 4: Each member of Huber's estimators, H_{ρ_1} , is minimal 2-feasible.

pf: See Appendix C.

The Tukey's estimator $\rho_2(\cdot, \cdot)$ (see Figure 2(b)) has zero weights⁵ for the large residues and thus improves the outlier rejection properties [4][5][11][14][19][31].

$$\rho_2(r, \sigma) = \begin{cases} \frac{\sigma^2}{6} \left[1 - \left(1 - \left(\frac{r}{\sigma} \right)^2 \right)^3 \right] & \text{if } r \leq \sigma, \\ \frac{\sigma^2}{6} & \text{otherwise.} \end{cases} \quad (8)$$

The shape of the function $\rho_2(\cdot, \cdot)$ is shown in Figure 2(b).

Property 5: Each member of Tukey's estimators, H_{ρ_2} , is minimal 2-feasible.

pf: See [6].

Another popular class of the robust error measure is the one proposed by Geman and McClure [12] (see Figure 2(c)) as shown below, which was also adopted in [3][4][23][31].

$$\rho_3(r, \sigma) = \frac{r^2}{r^2 + \sigma^2} \quad (9)$$

⁵ In robust statistics, the weight is defined to be an value proportional to the derivatives of ρ .

Property 6: Each member of Geman and McClures' estimators, H_{ρ_3} , is minimal 2-feasible.

pf: See [6].

The above three robust error measures are popular, and we have shown that all members of their H_{ρ} s have the minimal feasible value as 2. In addition, the following three robust error measures are also investigated.

The robust function $\rho_4(\cdot, \cdot)$ [36] (see Figure 2(d)) uses simple truncations to remove outliers, as shown in the following:

$$\rho_4(r, \sigma) = \begin{cases} r & \text{if } r \leq \sigma, \\ \sigma & \text{otherwise.} \end{cases} \quad (10)$$

Another two robust error measures investigated here are Lorentzian's estimator $\rho_5(\cdot, \cdot)$ [19] (see Figure 2(e)) and the trimmed mean M-estimator $\rho_6(\cdot, \cdot)$ [31] (see Figure 2(f)):

$$\rho_5(r, \sigma) = \log \left(1 + \frac{1}{2} \left(\frac{r}{\sigma} \right)^2 \right) \quad (11)$$

$$\rho_6(r, \sigma) = \begin{cases} \frac{r^2}{2} & \text{if } r \leq \sigma, \\ \frac{\sigma^2}{2} & \text{otherwise.} \end{cases} \quad (12)$$

Property 7: (i) Each member of H_{ρ_4} is minimal 1-feasible. (ii) Each member of H_{ρ_5} is minimal 2-feasible. (iii) Each member of H_{ρ_6} is minimal 2-feasible.

pf: See [6].

Notice that all the minimal feasible values of the commonly used non-decreasing robust error measures investigated above are 1 or 2, which are indeed small values.

In the following, we investigate an M-estimator that is not non-decreasing, the triweight M-estimator $\rho_7(\cdot, \cdot)$ [32]:

$$\rho_7(r, \sigma) = \begin{cases} r^2 & \text{if } r \leq \sigma, \\ \sigma^2 & \text{if } \sigma < r \leq 3\sigma, \\ 0 & \text{otherwise.} \end{cases} \quad (13)$$

Since each member of H_{ρ_7} is not non-decreasing, the process of template matching can not be speeded up with our approach if ρ_7 is selected to be the robust error measure.

Finally, we investigate the non-decreasing error measure shown as follows, where $k \in \mathbb{N}^+$.

$$\rho_8(r, \sigma) = r^k, \quad (14)$$

Property 8: Each member of H_{ρ_8} is minimal k -feasible.

pf: Trivial.

When $k=1$ (or $k=2$), ρ_8 becomes the SAD (or SSD) error measure. The SAD and SSD are therefore minimal 1-feasible and minimal 2-feasible, respectively. Hence, our method can also be used to speed up the template matching process where SAD or SSD is used as the error measure.

III. Search Strategy and The Main Algorithm

A. Search Strategy

Once an ascending lower bound list of the matching error is available for every search position (u, v) , many search strategies [26][25][7] can be used to speed up the process of robust template matching in our work. A brief review of these search strategies is given below. Without loss of generality, the p -pyramid serves as the pyramidal structure for describing these methods. We refer to p -pyramid as *pyramid* in the following. Consider a template I_t and a set of image blocks $F_{u,v}$ ($-W \leq u, v \leq W$) to be matched, where I_t and $F_{u,v}$ are both $N \times N$ images. Assume that the associated p -pyramids have been constructed for I_t and all of the $F_{u,v}$, $-W \leq u, v \leq W$, respectively.

In [26], only the highest level and the lowest level of the pyramids were used. The search order of the matching process is fixed and, without loss of generality, assume that the matching process starts from $F_{-W, -W}$ and the search is performed in a row-major order. First, the error $SRD_{\tau}(F_{-W, -W}, I_t)$ is computed as a reference value, r . Then, assume that (u', v') is the next site to be visited in the matching order. We try to find out whether the robust error measure $SRD_{\tau}(F_{u', v'}, I_t)$ is smaller than r . We do not compute $SRD_{\tau}(F_{u', v'}, I_t)$ directly. Instead, we first compute $SRD_{\tau}^{0,p}(F_{u', v'}, I_t)$, the error associated with the highest levels of the pyramids of I_t and $F_{u', v'}$. If $SRD_{\tau}^{0,p}(F_{u', v'}, I_t)$ is larger than the current reference value r , we do not have to further compute $SRD_{\tau}(F_{u', v'}, I_t)$ because $SRD_{\tau}^{0,p}(F_{u', v'}, I_t)$ is a lower bound of $SRD_{\tau}(F_{u', v'}, I_t)$. Therefore, early in the process we can jump out of the process of matching I_t and $F_{u', v'}$ and go on to match the next image block in the row-major

order. On the other hand, if $SRD_{\tau}^{0,p}(F_{u',v'}, I_t)$ is smaller than the current reference value r , we must compute $SRD_{\tau}(F_{u',v'}, I_t)$ and compare it with r . If $SRD_{\tau}(F_{u',v'}, I_t)$ is smaller than r , the current reference value, then r is replaced by $SRD_{\tau}(F_{u',v'}, I_t)$. The above procedure can be repeated iteratively. Remember that the computation complexity of $SRD_{\tau}^{0,p}(F_{u',v'}, I_t)$ is $\frac{1}{4^n}$ of that required for $SRD_{\tau}(F_{u',v'}, I_t)$, and thus such an early jump-out effect saves considerable computation time.

Lee and Chen [25] extended the idea of [26], using not only the highest and the lowest levels, but all the levels of the pyramids. First, the error $SRD_{\tau}(F_{-W,-W}, I_t)$ is also computed as a reference value, r . Then, once we begin to match I_t and $F_{u',v'}$ for some (u',v') depicted above, not only is $SRD_{\tau}^{0,p}(F_{u',v'}, I_t)$ computed, but a set of increasingly larger lower bounds, $SRD_{\tau}^{0,p}(F_{u',v'}, I_t)$, $SRD_{\tau}^{1,p}(F_{u',v'}, I_t)$, $SRD_{\tau}^{2,p}(F_{u',v'}, I_t)$, ..., are also computed in turn if necessary. Once some $SRD_{\tau}^{i,p}(F_{u',v'}, I_t)$ ($i = 0, 1, \dots, n$) is larger than the current reference value r , we have no need to compute all the other $SRD_{\tau}^{j,p}(F_{u',v'}, I_t)$ for $j = i+1, \dots, n$, and the matching process between I_t and $F_{u',v'}$ can be terminated. We can then jump to another matching process between I_t and the next image block of $F_{u',v'}$ in the row-major order. This method can be treated as using a depth-first search procedure in visiting the search tree as shown in Figure 3, and pruning the search branches once the computed error associated with the tree vertex is larger than the current reference value.

Recently, Chen et al. [7] extended the above method by exploiting the uniform cost search [34] in the tree instead of the depth-first search, so that the search order is not fixed. First, the smallest value among all the errors of the highest level is found:

$$SRD_{\tau}^{0,p}(F_{u^*,v^*}, I_t) = \min(\text{the elements of } A),$$

where $A = \{SRD_{\tau}^{0,p}(F_{-W,-W}, I_t), SRD_{\tau}^{0,p}(F_{-W,-W+1}, I_t), \dots, SRD_{\tau}^{0,p}(F_{W,W}, I_t)\}$ is referred to as the *active list*, and (u^*,v^*) is referred to as the temporary winner. Then, the error of the temporary winner in its next lower layer, $SRD_{\tau}^{1,p}(F_{u^*,v^*}, I_t)$, is computed. Next, the active list is updated by replacing $SRD_{\tau}^{0,p}(F_{u^*,v^*}, I_t)$ with $SRD_{\tau}^{1,p}(F_{u^*,v^*}, I_t)$:

$$A \leftarrow A \cup \{SRD_{\tau}^{1,p}(F_{u^*,v^*}, I_t)\} \setminus \{SRD_{\tau}^{0,p}(F_{u^*,v^*}, I_t)\}, \text{ where “}\setminus\text{” is the set difference.}$$

Then, the new minimal value among the elements in the new active list can be found:

$$SRD_{\tau}^{i,p}(F_{u_i^*,v_i^*}, I_t) = \min(\text{the elements of } A)$$

where i is now either 0 or 1, and a new temporary winner, (u^*,v^*) is obtained. Repeat the above procedure of alternately updating the active list and finding the minimal value of the elements

contained in it. Then, the minimal matching error can be found when (u^*, v^*) reaches the lowest level. In general, this method can prune more unnecessary branches because the uniform cost search strategy is used.

Once the p -pyramids have been constructed for the template and image blocks using the method introduced in this paper, it can be incorporated into any of the search strategies introduced above in order to speed up the process of robust template matching. In this work, the search strategy developed in [7] is adopted because the experimental results in [7] show that better speedup performance can be obtained compared with other approaches for non-robust template matching.

B. Main Algorithm

The algorithm of our approach using the uniform cost search for fast robust template matching is given below:

- Step 1. Initially, set $A = \{SRD_{\tau}^{0,p}(F_{-W,-W}, I_t), SRD_{\tau}^{0,p}(F_{-W,-W+1}, I_t), \dots, SRD_{\tau}^{0,p}(F_{W,W}, I_t)\}$.
- Step 2. Find (u^*, v^*) such that $SRD_{\tau}^{0,p}(F_{u^*,v^*}, I_t)$ is the minimum among all the elements contained in A .
- Step 3. $i \leftarrow 0$.
- Step 4. **While** ($i \neq n$)
- 4.1. Compute $SRD_{\tau}^{i+1,p}(F_{u^*,v^*}, I_t)$.
 - 4.2. $A \leftarrow A \cup \{SRD_{\tau}^{i+1,p}(F_{u^*,v^*}, I_t)\} \setminus \{SRD_{\tau}^{i,p}(F_{u^*,v^*}, I_t)\}$
 - 4.3. Find (u^*, v^*, j) such that $SRD_{\tau}^{j,p}(F_{u^*,v^*}, I_t)$ is the minimum among all the elements contained in A .
 - 4.4. $i \leftarrow j$.
- end While**
- Step 5. Output (u^*, v^*) .

The above algorithm applies the ‘‘uniform cost search’’ [34] to the tree illustrated in Figure 3, which guarantees to find the global minimum solution as demonstrated in the following. When the algorithm goes to step 5, we know that $SRD_{\tau}^{n,p}(F_{u^*,v^*}, I_t)$ is the minimum among all the elements contained in A . In addition, from Theorem 1, the matching error computed for any two images on the level n is not smaller than that for the other levels. Assume that $A = \{SRD_{\tau}^{n-w,-w,p}(F_{-W,-W}, I_t), SRD_{\tau}^{n-w,-w+1,p}(F_{-W,-W+1}, I_t), \dots, SRD_{\tau}^{n,w,w,p}(F_{W,W}, I_t)\}$. Then, $SRD_{\tau}^{n,p}(F_{i,j}, I_t) \geq SRD_{\tau}^{n_i,j,p}(F_{i,j}, I_t) \geq SRD_{\tau}^{n,p}(F_{u^*,v^*}, I_t)$ for all $SRD_{\tau}^{n_i,j,p}(F_{i,j}, I_t) \in A$, or equivalently, for all $-W \leq i, j \leq W$, which shows that

$SRD_{\tau}^{n,p}(F_{u^*,v^*}, I_t)$ is the global minimal on the level n .

In fact, instead of from the highest to the lowest levels (i.e., from level 0 to level n), the search process can also be performed from an arbitrary middle level, say m ($0 < m < n$), to the lowest level, n . This can simply be achieved by replacing each $SRD_{\tau}^{0,p}$ with $SRD_{\tau}^{m,p}$ in the Steps 1 and 2 of the main algorithm, and further modifying Step 3 to be $i = m$. However, in our experience, it is better to select the starting level based on the ratio of outliers. In particular, we find that starting from a middle level (instead of the highest level) usually makes the process of robust template matching more efficient in practice. This is because the outliers contained in the template are included in the highest level image, although they may not be included in some middle level images. Some experimental results for the speedup versus different combinations of the starting levels and outlier ratios are shown in Section IV.

In practice, some standard methods [30][38][40] can be used to estimate an appropriate value for the parameter σ in robust estimation. For example, the ‘‘median absolute deviation’’ scale estimate, which is related to the median of the absolute values of the residuals, is given by

$$\hat{\sigma} = 1.4826 \left[1 + \frac{5}{(n-p)} \right] \text{median}_i |r_i|, \quad (15)$$

where the constant 1.4826 is a finite sample correction factor, n is the size of the data set, p is the dimension of the parameter vector, and r_i is the residual error.

IV. Experimental Results

In this section, we present the results of three different experiments, including signal matching, face template matching, and motion estimation.

A. Signal Matching

In this experiment, we perform a simulation of searching a particular 1-D pattern along a 8192-point input signal, ranging from 0 to 255, which is synthesized using a linear regression model. Four such input signals are used for this experiment, and one of them is shown in Figure 4(a). We first randomly extract a 512-point partial segment, which is called the *true identity signal*, from an input signal. A 512-point test signal can then be generated by adding both Gaussian noise and some outliers to the true identity signal, as shown in Figures 4(b), 4(c), and 4(d), respectively. The outlier ratio (i.e., the ratio of the number of outlier points to the length of a test signal, 512) varies from 0 to 0.15. Then the test signal is used as a template and we try to find its matching segment in the input

signal from which the test signal is extracted. In fact, this experiment simulates stereo matching (i.e., matching along a scan-line or epi-polar line) in computer vision [20]. Here, the SRD is used as the matching criterion. For each input signal and each outlier ratio, we first randomly generate 30 test signals and then find their matching segment in the input signal. The simple-truncation function (10) is used in this experiment as a robust error measure to suppress the effects of outliers. If the matching segment is not equal to its true identity signal, a miss occurs. Otherwise, a hit occurs. In this experiment, high average hit ratios ranging from 99.7% to 99.8% are achieved for all outlier ratios tested, which indicates that the M-estimator is very useful for suppressing the outlier effects. In the following, we focus on the main issue of this paper, the speedup of robust template matching, by comparing the efficiencies of our method with respect to those of the FS method.

First, we investigate the advantage of our method for the reduction of the major operations involved in the SRD computations. We define a *robust operation* to be the computation of the robust error measure $\rho(\cdot, \cdot)$, and computing $\rho(\cdot, \cdot)$ n times is therefore referred to as that n robust operations are performed. The robust operation is generally the most critical part for obtaining an SRD. The efficiency improvement with our method is evaluated by comparing the following two ratios: (1) the ratio of robust operations involved in our method to that in the FS method, and (2) the ratio of the execution time with our method to that with the FS method. These two ratios are referred to as the *operation count ratio* and the *time consumption ratio*, respectively. Evaluation of the efficiency improvement based on the operation count ratios is machine independent, but some additional computational overheads such as the construction of the p -pyramids and the switching among the search branches can not be reflected by the operation count ratios. On the other hand, evaluation based on the time consumption ratios includes all the overheads, but is machine dependent. In this experiment, both the operation count ratios and time consumption ratios are computed for evaluations and comparisons. In the setting of this experiment, there are a total of 10 levels (level 0 to level 9) in the pyramid because the length of the test signal, 512, is equal to 2^9 . Remember that our algorithm can start from any of the middle levels, as described in Section IV. Therefore, we also compare the speedup effect when the matching processes start from different initial levels in this experiment.

Since the simple truncation function is minimal 1-feasible, the 1-pyramid is constructed for robust template matching with our method. The operation count ratios using 1-pyramid are shown in Figure 5(a). From Figure 5(a), the operation count ratio varies overall from 0.1 to 0.48 when the outlier ratio varies from 0 to 0.15, if the middle value of the operation count ratios serves as a representative for each outlier ratio. This shows that our method can successively reduce the number of the major operations required for robust template matching, and the smaller the outlier ratio, the more reductions are achieved. An interesting phenomenon is that the best starting level (i.e., the starting level associated with the smallest operation count ratio) for each outlier ratio is a middle level.

For example, when the outlier ratio lies in $[0, 0.02]$, the best starting level is 5; whereas when the outlier ratios are increased to be within $[0.04, 0.08]$ and $[0.1, 0.15]$, the associated best starting levels become 6 and 7, respectively. The reason for this is that the outliers are easily to be included in the accumulation process for building a high level image in the pyramid, and thus almost all the lower bounds with respect to a high level image are required to be computed in the search process. On the other hand, when starting from a low level image, the lower bounds may not be tight enough to prune the search branches. Another interesting phenomenon is that the best starting level becomes lower as the outlier ratio increases.

Figure 5(b) shows the time-consumption ratios taking into account not only the major operations but all the computational overheads such as pyramid constructions and controlling processes of search. The test was performed on a PC with the Visual C++ language, and the middle levels, 5, 6, 7, and 8 were used as the starting levels, respectively. From Figure 5(b), the time consumption ratio varies overall from 0.4 to 0.6 when the outlier ratio varies from 0 to 0.15, indicating that our method can also increase the efficiency of robust template matching in practice. Similarly, the best starting level becomes lower when the outlier ratio increases, and the best time consumption ratio varies from 0.2 to 0.4. Hence, if priori knowledge about the outlier ratio of the template matching problem to be solved is given in advance, this can serve as a guideline to choose the best (or a better) starting level.

Notice that the simple truncation is minimal 1-feasible, and it is therefore also p -feasible for all $p \in [1, \infty]$. In the following, different p -pyramids (1-pyramid, 2-pyramid, and ∞ -pyramid) are respectively constructed to compare their efficiencies when they are incorporated into our method. Their average operation count and time consumption ratios are summarized in Table 1, showing that the speedup performance degrades when either the 2-pyramid or the ∞ -pyramid is used. This matches our claim above that the closer is p to its minimal feasible value, the better speedup is achieved when the p -pyramid is used.

A summary of the above experimental results is given below. First, the hit ratio is high when the technique of robust template matching is used, confirming that the SRD can suppress the affection of outliers. Second, from both the operation count and the time consumption ratios, our method is more efficient than the FS method. The amount of speedup achieved depends on many issues such as outlier ratios, starting levels, and the p -pyramid being used. In practice, the starting level should be selected according to the outlier ratio. If an estimation of the outlier is available in advance, the best starting level can be chosen according to the corresponding simulation results. As for which p -pyramid is suitable to be adopted in our method for a given robust error measure, it is suggested that the closer is p to its minimal feasible value, the more speedup is achieved.

Table 1. Comparisons of average time consumption ratio (tc) and average operation count ratios (oc) of our method when 1-pyramid, 2-pyramid, and ∞ -pyramid are used.

		1-pyramid		2-pyramid		∞ - pyramid	
		oc	tc	oc	tc	oc	tc
outlier ratio 0~0.4	start from level 8	0.5	0.755	0.5	0.716	0.506	0.696
	start from level 7	0.25	0.441	0.256	0.436	0.291	0.468
	start from level 6	0.13	0.268	0.142	0.297	0.259	0.474
	start from level 5	0.102	0.250	0.139	0.331	0.314	0.629
outlier ratio 0.5~0.9	start from level 8	0.5	0.735	0.501	0.715	0.525	0.692
	start from level 7	0.25	0.419	0.269	0.443	0.392	0.595
	start from level 6	0.159	0.292	0.246	0.449	0.462	0.76
	start from level 5	0.22	0.458	0.309	0.619	0.536	0.951
outlier ratio 0.10~0.15	start from level 8	0.5	0.714	0.507	0.707	0.588	0.753
	start from level 7	0.255	0.381	0.353	0.541	0.588	0.828
	start from level 6	0.365	0.582	0.470	0.758	0.730	1.081
	start from level 5	0.429	0.737	0.533	0.904	0.796	1.234

B. Face Template Matching

We perform face template matching experiments in a face-only database [27], which can be used for the application of finding a particular person in a database. We use 1000 images of 100 persons, where each person has 10 images with distinct poses or expressions per person. Each image size is normalized to be 64x64 and part of this subset is shown in Figure 6(a). For each person, we randomly select one of his (or her) images to be the test image, and the other 9 images remain in the database. Hence, there are a total of 900 images contained in the database in our experiment. All test images are contaminated by pepper-and-salt noise that is used as outliers, with the outlier ratios varying from 0 to 0.15. Figure 6(b) shows the contaminated images of a person with different outlier ratios. Such an experimental setup is similar to that shown in [24], but in our case the templates are polluted with outliers. Given a test image, we match it with the 900 images contained in the database and find the most similar one with the least sum of robust differences. The person with respect to the most similar image is then served as the recognized one. The matching experiment is performed for all of the 100 test images, and the average recognition rate (i.e. the hit ratio) is recorded.

Four different robust estimators, Huber's estimator, Tukey's estimator, Geman and McClure's estimator, and the trimmed mean M-estimator, are used in this experiment. Based on (15), the parameter σ used in these estimators was determined by performing several random matches to

estimate $\text{median}_i |r_i|$ in advance. The associated 2-pyramids are constructed based on these robust estimators to speed up the corresponding robust template matching processes. First, we compare the recognition performances of template matching, using the above four robust estimators to that using the SSD criterion. Figures 7(a), 8(a), 9(a), and 10(a) show the comparison results for Huber's estimator, Tukey's estimator, Geman and McClure's estimator, and the trimmed mean M-estimator, respectively. From these figures, the hit ratios obtained using SRD are generally better than those obtained using SSD, no matter which estimators are used. In particular, the hit ratios obtained by using Tukey's estimator and Geman and McClure's estimators consistently perform better than those using SSD in all experiments, no matter which outlier ratios are tested. This observation also confirms that the M-estimator can deal with outliers better.

In the following, we present the speedup performances of our method. In particular, we focus on the time consumption ratios in this experiment. Since the pyramids of the images contained in the database are constructed offline for this application, the pyramid-construction time for the images contained in the database is not included in the time consumption ratio in this experiment. However, note that the pyramid construction time of the test image has remained to be included in the computation of time consumption ratios. The time consumption ratios of the above four robust estimators are shown in Figures 7(b), 8(b), 9(b), and 10(b), respectively. From these figures, it can be seen that our method can increase the efficiencies for the face template matching for different kinds of robust estimators. In essence, the smaller is the outlier ratio, the better speedup is achieved. In particular, the speed performances depend on what kind of robust estimators are used. For example, the speedup performances are better when Huber's estimator, Tukey's estimator, and the trimmed mean M-estimator are used, than that when Geman and McClure's estimator is used. A possible reason for this is that the lower bounds derived for the former estimators are tighter than those for the latter.

C. Motion Estimation

In the last experiment, we use our method for robust motion estimation in a sequence of images. Each image in the sequence is segmented into a set of blocks, and we try to find the motion vector for each block. Assume that one image in the sequence is polluted with outliers, and the robust template matching technique is used for motion estimation when outliers occur. The Salesman image sequence is used as a test sequence where each frame is of size 352×288 . Two different block sizes (16×16 and 32×32) and their corresponding search ranges (about twice as large as the block size, $[-16, 16] \times [-16, 16]$ and $[-32, 32] \times [-32, 32]$) are tested in this experiment. Given a pair of two consecutive frames, where one is polluted as shown in Figure 11, we use the simple truncation (10) as the robust error measure for robust template matching to find the motion vectors. The time

consumption ratios (that consider all the computational overheads) and the operation count ratios are shown in Figure 12, respectively. The experimental results show that our method can also improve the efficiency for robust motion estimation.

V. Discussion

From the experimental results, the efficiency improvement of our method improves if the outlier ratio gets smaller. In fact, the time-consumption ratio may exceed a value of one in our experiment when the outlier ratio is too large, which means that our method cannot speed up the matching process in this case. However, since the estimation problem is itself more difficult to solve accurately when the outlier ratio is too large, our method can be treated as dealing with common cases of robust template matching when M-estimators are used.

One possible means to allow large outlier ratios is to make the lower bounds tighter. In fact, the set of ascending lower bounds (accompanied with the minimal feasible value and the p -pyramid structure) derived in this paper is generally available for all non-decreasing robust error measures. Improvement is thus possible by tightening the lower bounds for a particular error measure, although the improved set of lower bounds may not be able to be used for other error measures. To achieve this goal, we can take advantage of the specific mathematical form of each of the error measures and derive new theoretical bounds and pyramid structures. Such an improvement remains an open problem and to be investigated in the future.

Our main algorithm was presented to tackle the matching problem with translation. When dealing with a problem with rotation or scaling, a useful strategy is to generate a set of templates by rotating and scaling the pattern to be matched in advance [21][37]. In this case, a p -pyramid can be produced for each template and our algorithm can then be applied for speeding up the matching process with these templates. Such a pre-transforming/pre-storing process is suitable for applications where offline processing is allowed, such as the face-matching task shown in the experimental results of this paper. Another way for dealing with rotation (or illumination variation) is to use rotationally invariant (or lighting-invariant) features such as moment invariants [10]. By this way, the templates need to be pre-processed to extract their invariant feature vectors. To our best knowledge, however, there is no analysis about the sensitivities of the invariant vectors to outliers, and thus it is possible that they are not robust enough for matching with outliers. Nevertheless, when the noise model is Gaussian and a common SSD measure is applied for matching the feature vectors, our method can also be used to speed up the matching process by building the 2-pyramids associated with the invariant feature vectors in advance.⁶ In addition, another widely-used way for dealing with

⁶ Note that extraction of moment invariants can itself be speeded up with some other fast algorithms [9].

illumination variation is to perform histogram equalization [18].

Another issue from the experimental results is that the best starting level becomes lower as the outlier ratio gets smaller. How to determine the best (or a better) starting level in an *analytic* way when the outlier ratio and a particular robust error measure are given also remains an open problem. In the meantime, the best starting level can be determined through a simulation process by taking some training examples from the problem to be solved.

VI. Conclusions

In this paper, we have developed a systematic method that can be used to speed up the process of robust template matching when the error function is non-decreasing. We proposed the p -feasible related theory in Section II, which serves as a basis for constructing a hierarchical structure, the p -pyramid. This pyramid can then be used to generate a set of ascending lower bounds of the minimal matching error. As long as the error function is non-decreasing, there exists a p making this function p -feasible. We have also shown that the smaller is p , the more efficient is the associated fast algorithm. In addition, many commonly used non-decreasing robust estimators are minimal 2-feasible or minimal 1-feasible, as shown in Section II.E.

By exploiting the property that the computations of the lower bounds generated with our method are more efficient than the computation of the matching errors directly, some fast search strategies can thus be used to speed up the matching process, as shown in Section III.A. In this paper, the uniform cost search strategy is adopted, and the experimental results show that our method can successfully increase the matching efficiencies when outliers exist.

A characteristic of our method is that it is easily generalized as not finding only the site with the minimal matching error, but a series of sites having the k minimal matching errors. To compute the k minimal matching errors, we do not need to perform our algorithm k times, but continuously execute the best-search procedure shown in Section III.B until k leaf nodes have been computed. Hence, the robust k -nearest neighbor problem can be solved in a unique procedure. Moreover, our method can also be used for improving the matching efficiency if a sub-optimal or approximate solution is allowed to be generated instead of the solution with the minimal matching error. This can be achieved by using a middle level, instead of the lowest level, as being the level containing the leaf nodes.

To our knowledge, this is the first approach proposed to speed up the process of robust template matching. The method proposed in this paper gives a generally useful scenario for solving this type

of problem. Our method also provides some significant research directions that can be further studied, including the investigations of tighter lower bounds, the generalizations to include the error measures that are not non-decreasing, and the analytical decisions for best starting levels.

Appendix A: Proof of Property 2

Proof: If a robust error measure $\tau(\cdot)$ is p -feasible, to prove that $\tau(\cdot)$ is also q -feasible for all $q \in [p, \infty]$ is equivalent to prove that $\tau(\cdot)$ satisfies the following inequality for all $q \in [p, \infty]$.

$$\forall a_1, a_2 \in \mathcal{R}^+ \cup \{0\}, \quad \tau(a_1) + \tau(a_2) \geq \tau(\|a\|_q) \quad (\text{A.1})$$

We first introduce three lemmas before proving (A.1) for all $q \in [p, \infty]$. In the following, assume that $\mathbf{x} = [x_1 \ x_2]^T \in \mathcal{R}^2$. In addition, we denote, $\mathbf{B}_{\|\cdot\|_\alpha} = \{\mathbf{x} \mid \|\mathbf{x}\|_\alpha \leq 1\}$, to be the unit ball with respect to the L_α norm.

Lemma 1. If $\alpha \in [1, \infty]$ and $\mathbf{x} \in \mathbf{B}_{\|\cdot\|_\alpha}$, then $1 \geq |x_1| \geq 0$ and $1 \geq |x_2| \geq 0$.

Proof: trivial.

Lemma 2. If $\alpha, \beta \in [1, \infty]$ and $\beta \geq \alpha$, then $\mathbf{B}_{\|\cdot\|_\alpha} \subseteq \mathbf{B}_{\|\cdot\|_\beta}$.

Proof: We prove this lemma by considering the following two cases, $\alpha < \beta$ and $\alpha = \beta$.

Case 1 ($\alpha < \beta$): $\forall \mathbf{x} \in \mathbf{B}_{\|\cdot\|_\alpha}$

$$\begin{aligned} & \left(|x_1|^\alpha + |x_2|^\alpha \right)^{\frac{1}{\alpha}} \leq 1 \\ & \Rightarrow 0 \leq \left(|x_1|^\alpha + |x_2|^\alpha \right) \leq 1 \\ & \Rightarrow 0 \leq \left(|x_1|^\beta + |x_2|^\beta \right) = \left(|x_1|^{\beta-\alpha} |x_1|^\alpha + |x_2|^{\beta-\alpha} |x_2|^\alpha \right) \leq \left(|x_1|^\alpha + |x_2|^\alpha \right) \leq 1 \\ & \Rightarrow 0 \leq \left(|x_1|^\beta + |x_2|^\beta \right)^{\frac{1}{\beta}} \leq 1 \\ & \Rightarrow \mathbf{x} \in \mathbf{B}_{\|\cdot\|_\beta} \end{aligned}$$

Case 2 ($\alpha = \beta$): trivial.

From the above two cases, if $\alpha, \beta \in [1, \infty]$ and $\beta \geq \alpha$, then $\mathbf{B}_{\|\cdot\|_\alpha} \subseteq \mathbf{B}_{\|\cdot\|_\beta}$.

Lemma 3. If $\alpha, \beta \in [1, \infty]$ and $\mathbf{B}_{\|\cdot\|_\alpha} \subset \mathbf{B}_{\|\cdot\|_\beta}$, then $\forall \mathbf{x}, \|\mathbf{x}\|_\alpha \geq \|\mathbf{x}\|_\beta$.

Proof: see p. 282 in [17].

From Lemma 2 and Lemma 3, we know that if $\alpha, \beta \in [1, \infty]$ and $\alpha \leq \beta$, then $\forall \mathbf{x}, \|\mathbf{x}\|_\alpha \geq \|\mathbf{x}\|_\beta$. Since $\tau(\cdot)$

is p -feasible and $\|\mathbf{x}\|_p \geq \|\mathbf{x}\|_q$ for all $q \in [p, \infty]$, then (A.1) holds for all $q \in [p, \infty]$. Hence, $\tau(\cdot)$ is also q -feasible for all $q \in [p, \infty]$.

Appendix B: Proof of Theorem 1

Proof: Given a robust error measure $\tau(\cdot)$ that is p -feasible, for all $m \in \{0, 1, \dots, n-1\}$,

$$\begin{aligned}
& SRD_{\tau}^{m+1,p}(I_1, I_2) \\
&= \sum_{0 \leq i, j < 2^{m+1}} \tau\left(I_1^{m+1,p}(i, j) - I_2^{m+1,p}(i, j)\right) \\
&= \sum_{0 \leq i, j < 2^m} \left\{ \tau\left(I_1^{m+1,p}(2i, 2j) - I_2^{m+1,p}(2i, 2j)\right) + \tau\left(I_1^{m+1,p}(2i, 2j+1) - I_2^{m+1,p}(2i, 2j+1)\right) \right. \\
&\quad \left. + \tau\left(I_1^{m+1,p}(2i+1, 2j) - I_2^{m+1,p}(2i+1, 2j)\right) + \tau\left(I_1^{m+1,p}(2i+1, 2j+1) - I_2^{m+1,p}(2i+1, 2j+1)\right) \right\} \\
&\geq \sum_{0 \leq i, j < 2^m} \left\{ \tau\left(\left(\left| I_1^{m+1,p}(2i, 2j) - I_2^{m+1,p}(2i, 2j) \right|^p + \left| I_1^{m+1,p}(2i, 2j+1) - I_2^{m+1,p}(2i, 2j+1) \right|^p \right)^{\frac{1}{p}} \right) \right. \\
&\quad \left. + \tau\left(\left(\left| I_1^{m+1,p}(2i+1, 2j) - I_2^{m+1,p}(2i+1, 2j) \right|^p + \left| I_1^{m+1,p}(2i+1, 2j+1) - I_2^{m+1,p}(2i+1, 2j+1) \right|^p \right)^{\frac{1}{p}} \right) \right\} \\
&\geq \sum_{0 \leq i, j < 2^m} \left\{ \tau\left(\left(\left| I_1^{m+1,p}(2i, 2j) - I_2^{m+1,p}(2i, 2j) \right|^p + \left| I_1^{m+1,p}(2i, 2j+1) - I_2^{m+1,p}(2i, 2j+1) \right|^p \right. \right. \right. \\
&\quad \left. \left. + \left| I_1^{m+1,p}(2i+1, 2j) - I_2^{m+1,p}(2i+1, 2j) \right|^p + \left| I_1^{m+1,p}(2i+1, 2j+1) - I_2^{m+1,p}(2i+1, 2j+1) \right|^p \right)^{\frac{1}{p}} \right) \right\} \\
&\geq \sum_{0 \leq i, j < 2^m} \left\{ \tau\left(\left(I_1^{m+1,p}(2i, 2j)^p + I_1^{m+1,p}(2i, 2j+1)^p + I_1^{m+1,p}(2i+1, 2j)^p + I_1^{m+1,p}(2i+1, 2j+1)^p \right)^{\frac{1}{p}} \right. \right. \\
&\quad \left. \left. - \left(I_2^{m+1,p}(2i, 2j)^p + I_2^{m+1,p}(2i, 2j+1)^p + I_2^{m+1,p}(2i+1, 2j)^p + I_2^{m+1,p}(2i+1, 2j+1)^p \right)^{\frac{1}{p}} \right) \right\} \\
&= SRD_{\tau}^{m,p}(I_1, I_2),
\end{aligned}$$

where the first two inequalities are based on constraint (4), and the last inequality is derived from both the triangle inequality⁷ and the fact that $\tau(\cdot)$ is non-decreasing.

Appendix C: Proof of Property 4

Assume that $\tau(\cdot)$ belongs to H_{ρ_1} . To prove that $\tau(\cdot)$ is minimal 2-feasible is equivalent to showing that $\tau(\cdot)$ is 2-feasible but not 1-feasible. Since it is trivial that $\tau(\cdot)$ is both non-decreasing and not 1-feasible, we focus on the proof of the fact that $\tau(a_1) + \tau(a_2) \geq \tau(\|\mathbf{a}\|_2)$ for all $a_1, a_2 \in \mathcal{R}^+ \cup \{0\}$. Without loss of generality, assume that $0 \leq a_2 \leq a_1 \leq \|\mathbf{a}\|_2$, and consider the following four cases:

⁷ Given two vectors \mathbf{a} and \mathbf{b} , $\|\mathbf{a}-\mathbf{b}\|_p \geq \|\mathbf{a}\|_p - \|\mathbf{b}\|_p$

- Case 1 ($\sigma \geq \|\mathbf{a}\|_2 \geq a_1 \geq a_2 \geq 0$):

$$\rho_1(a_1, \sigma) + \rho_1(a_2, \sigma) - \rho_1(\|\mathbf{a}\|_2, \sigma) = \frac{a_1^2}{2} + \frac{a_2^2}{2} - \frac{a_1^2 + a_2^2}{2} = 0 \quad (\text{C.1})$$

- Case 2 ($\|\mathbf{a}\|_2 \geq \sigma \geq a_1 \geq a_2 \geq 0$):

$$\rho_1(a_1, \sigma) + \rho_1(a_2, \sigma) - \rho_1(\|\mathbf{a}\|_2, \sigma) = \frac{a_1^2}{2} + \frac{a_2^2}{2} - \sigma\sqrt{a_1^2 + a_2^2} + \frac{\sigma^2}{2}. \quad (\text{C.2})$$

Since $\frac{(a_1^2 + a_2^2) + \sigma^2}{2} \geq \sigma\sqrt{a_1^2 + a_2^2} \geq 0$ according to the arithmetic-geometric inequality, (C.2)

≥ 0 .

- Case 3 ($\|\mathbf{a}\|_2 \geq a_1 \geq \sigma \geq a_2 \geq 0$):

$$\rho_1(a_1, \sigma) + \rho_1(a_2, \sigma) - \rho_1(\|\mathbf{a}\|_2, \sigma) = \sigma a_1 + \frac{a_2^2}{2} - \sigma\sqrt{a_1^2 + a_2^2} \quad (\text{C.3})$$

Since $\left(\sigma a_1 + \frac{a_2^2}{2}\right)^2 - \left(\sigma\sqrt{a_1^2 + a_2^2}\right)^2 = \sigma a_1 a_2^2 + \frac{a_2^4}{4} - \sigma^2 a_2^2 = a_2^2 \left(\sigma(a_1 - \sigma) + \frac{a_2^2}{4}\right) \geq 0$ and both two

terms $\sigma a_1 + \frac{a_2^2}{2}$ and $\sigma\sqrt{a_1^2 + a_2^2}$ are nonnegative, (C.3) ≥ 0 .

- Case 4 ($\|\mathbf{a}\|_2 \geq a_1 \geq a_2 \geq \sigma > 0$):

$$\rho_1(a_1, \sigma) + \rho_1(a_2, \sigma) - \rho_1(\|\mathbf{a}\|_2, \sigma) = \sigma \left(a_1 + a_2 - \sqrt{a_1^2 + a_2^2} - \frac{\sigma}{2} \right) \quad (\text{C.4})$$

Since $\left(a_1 + a_2 - \frac{\sigma}{2}\right)^2 - \left(\sqrt{a_1^2 + a_2^2}\right)^2 = 2a_1 a_2 - \sigma a_1 - \sigma a_2 + \frac{\sigma^2}{4} = a_1(a_2 - \sigma) + a_2(a_1 - \sigma) + \frac{\sigma^2}{4} > 0$ and

both two terms $a_1 + a_2 - \frac{\sigma}{2}$ and $\sqrt{a_1^2 + a_2^2}$ are nonnegative, then (C.4) ≥ 0 .

The union of the above four cases shows that $\tau(a_1) + \tau(a_2) \geq \tau(\|\mathbf{a}\|_2)$ for all $a_1, a_2 \in \mathbb{R}^+ \cup \{0\}$.

References

- [1] M. Bierling, "Displacement estimation by hierarchical block-matching," in *Proceedings of SPIE Conference on Visual Communications and Image Processing*, vol. 1001, pp. 942-951, 1988.
- [2] V. Barnett and T. Lewis, *Outliers in Statistical Data*. New York: John Wiley and Sons, 1994.
- [3] M. J. Black and P. Anandan, "The robust estimation of multiple motions: parametric and piecewise-smooth flow fields," *Computer Vision and Image Understanding*, vol. 63, pp. 75-104, 1996.
- [4] M. J. Black and A. Rangarajan, "On the unification of line processes, outlier rejection, and robust statistics with applications in early vision," *International Journal of Computer Vision*, vol. 19, pp. 57-91, 1996.

- [5] A. Can, C. V. Stewart, and B. Roysam, "Robust hierarchical algorithm for constructing a mosaic from images of the curved human retina," in *Proceedings of IEEE Conference Computer Vision and Pattern Recognition*, pp. 286-292, 1999.
- [6] J.-H. Chen, C.-S. Chen, and Y.-S. Chen, "Fast method for robust template matching," *Technical Report TR-IIS-01-015*, Institute of Information Science, Academia Sinica, Taipei, Taiwan, 2001.
- [7] Y.-S. Chen, Y.-P. Hung, and C.-S. Fuh, "Fast block matching algorithm based on the winner-update strategy," *IEEE Transactions on Image Processing*, vol. 10, pp. 1212-1222, 2001.
- [8] K. H. K. Chow and M. L. Liou, "Genetic motion search algorithm for video compression," *IEEE Transactions on Circuit System and Video Technology*, vol. 3, pp. 440-445, 1993.
- [9] J. Flusser, "Refined moment calculation using image block representation," *IEEE Transactions on Image Processing*, vol. 11, pp. 1977-1978, 2000.
- [10] J. Flusser and B. Zitová, "Combined invariants to linear filtering and rotation," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 13, pp. 1123-1136, 1999.
- [11] H. Frigui and R. Krishnapuram, "A robust competitive clustering algorithm with applications in computer vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, to appear.
- [12] S. Geman and D. E. McClure, "Statistical methods for tomographic image reconstruction," *Bulletin of the International Statistical Institute*, vol. 52, pp.5-21, 1987.
- [13] M. Ghanbari, "The cross-search algorithm for motion estimation," *IEEE Transactions on Communication*, vol. 38, pp. 950-953, 1990.
- [14] R. M. Haralick, H. Joo, C.-N. Lee, and et al., "Pose estimation from corresponding point data," *IEEE Transactions Systems, Man, and Cybernetics*, vol. 19, pp. 1426-1446, 1989.
- [15] H.-C. Huang, Y.-P. Hung, and W.-L. Hwang, "Adaptive early jump-out technique for fast motion estimation in video coding," *CVGIP: Graphical Models and Image Processing*, vol. 59, pp. 388-394, 1997.
- [16] P. J. Huber, *Robust Statistics*. New York: John Wiley and Sons, 1981.
- [17] R. A. Horn, and C. R. Johnson, *Matrix Analysis*. Cambridge University Press, 1985.
- [18] K. Jonsson, J. Kittler, Y. P. Li, and J. Matas, "Support vector machines for face authentication," *Image and Vision Computing*, to appear.
- [19] K. Kawamura, K. Hasegawa, O. Yamashita, and et al., "Object recognition using local EGI and 3D models with M-estimators," in *Proceedings of IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pp. 80-86, 1999.
- [20] R. Klette, K. Schlüns, and A. Koschan, *Computer Vision Three-Dimensional Data from Images*, Chapter 4, Springer, 1998.
- [21] D. Keysers, W. Macherey, J. Dahmen, and H. Ney. "Learning of Variability for Invariant Statistical Pattern Recognition," in *Proceedings of 12th European Conference on Machine Learning*, Freiburg, Germany, pp. 263-275, 2001.
- [22] T. Koga, K. Iinuma, A. Hirano, and et al., "Motion compensated interframe coding for video conference," in *Proceedings of National Telecommunications Conference*, pp. G5.3.1-G5.3.5, 1981.
- [23] S.-H. Lai, "Robust image matching under partial occlusion and spatially varying illumination change," *Computer Vision and Image Understanding*, vol. 78, pp. 84-98, 2000.
- [24] S. Lawrence, C. L. Giles, A. -C. Tsoi, and A. Back, "Face recognition: a convolutional neural network approach," *IEEE Transactions on Neural Networks*, vol. 8, pp. 98-113, 1997.
- [25] C.-H. Lee and L.-H. Chen, "A fast motion estimation algorithm based on the block sum pyramid," *IEEE Transactions on Image Processing*, vol. 6, pp. 1587-1591, 1997.

- [26] W. Li and E. Salari, "Successive elimination algorithm for motion estimation," *IEEE Transactions on Image Processing*, vol. 4, pp. 105-107, 1995.
- [27] H.-Y. M. Liao, C.-C. Han, and G.-J. Yu, "Face+hair+shoulders+background \neq face," *Technical Report TR-IIS-97-007*, Institute of Information Science, Academia Sinica, Taipei, Taiwan, 1997.
- [28] C.-H. Lin and J.-L. Wu, "A lightweight genetic block-matching algorithm for video coding," *IEEE Transactions on Circuit System Video Technology*, vol. 8, pp. 386-392, 1998.
- [29] L.-K. Liu and E. Feig, "A block-based gradient descent search algorithm for block motion estimation in video coding," *IEEE Transactions on Circuit System Video Technology*, vol. 6, pp. 419-422, 1996.
- [30] P. Meer, D. Mintz, A. Rosenfeld and D. Y. Kim, "Robust regression methods for computer vision: a review," *International Journal of Computer Vision*, vol. 6, pp. 59-70, 1991.
- [31] M. J. Mirza and K. L. Boyer, "Performance evaluation of a class of M-estimators for surface parameter estimation in noisy range data," *IEEE Transactions on Robotics and Automation*, vol. 9, pp. 75-85, 1993.
- [32] S. I. Olsen, "Epipolar line estimation," in *Proceedings of Second European Conference on Computer Vision*, pp. 307-311, 1992.
- [33] J. Park, B. Jiang, and U. Neumann, "Vision-based pose computation: robust and accurate augmented reality tracking," in *Proceedings of the IEEE International Workshop on Augmented Reality*, pp. 3-12, 1999.
- [34] S. Russel and P. Norvig, *Artificial Intelligence*, pp. 75, Prentice Hall, New Jersey, 1995.
- [35] M.-C. Shie, W.-H. Fang, K.-J. Hung, and et al., "Fast, robust block motion estimation using simulated annealing," *IEICE Transactions on Fundamentals*, vol. E83, pp. 121-127, 2000.
- [36] D.-G. Sim, O.-K. Kwon, and R.-H. Park, "Object matching algorithms using robust hausdorff distance measures," *IEEE Transactions Image Processing*, vol. 8, pp. 425-429, 1999.
- [37] P. Simard, Y. Le Cun, J. Denker, "Efficient Pattern Recognition Using a New Transformation Distance", in *Neural Information Processing Systems*, S. J. Hanson, J. D. Cowan, and C. L. Giles, Eds. 1993, vol. 5, pp. 50-58, Morgan Kaufmann.
- [38] C. V. Stewart, "Robust parameter estimation in computer vision," *SIAM Review*, vol. 41, pp. 513-537, 1999.
- [39] S. Zhu and K.-K. Ma, "A new diamond search algorithm for fast block-matching motion estimation," *IEEE Transactions Circuit System and Video Technology*, vol. 9, pp. 287-290, 2000.
- [40] Z. Zhang, "Parameter estimation techniques: a tutorial with application to conic fitting," *Image and Vision Computing*, Vol. 15, pp. 59-76, 1997.

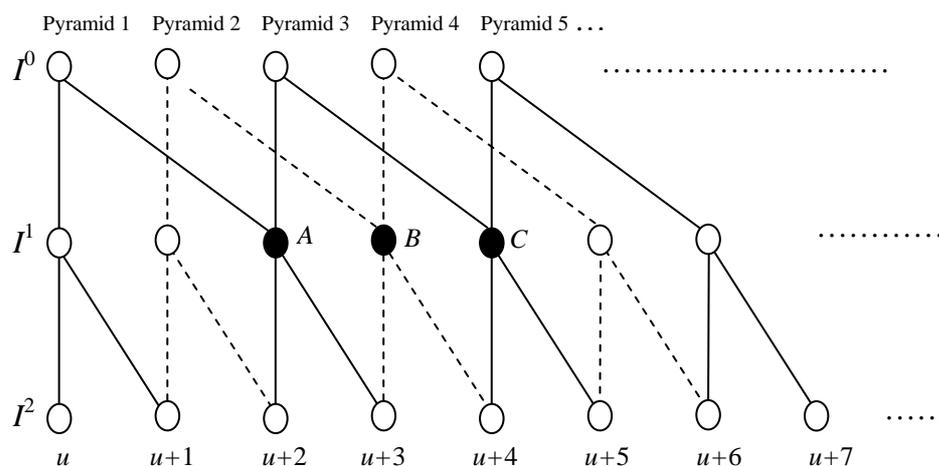


Figure 1. This figure demonstrates the p -pyramid constructed from a 1-d signal where each element in higher levels is composed of its two son elements. Five p -pyramids are depicted in solid or dashed lines. The three black nodes are the ones that are shared between two pyramids.

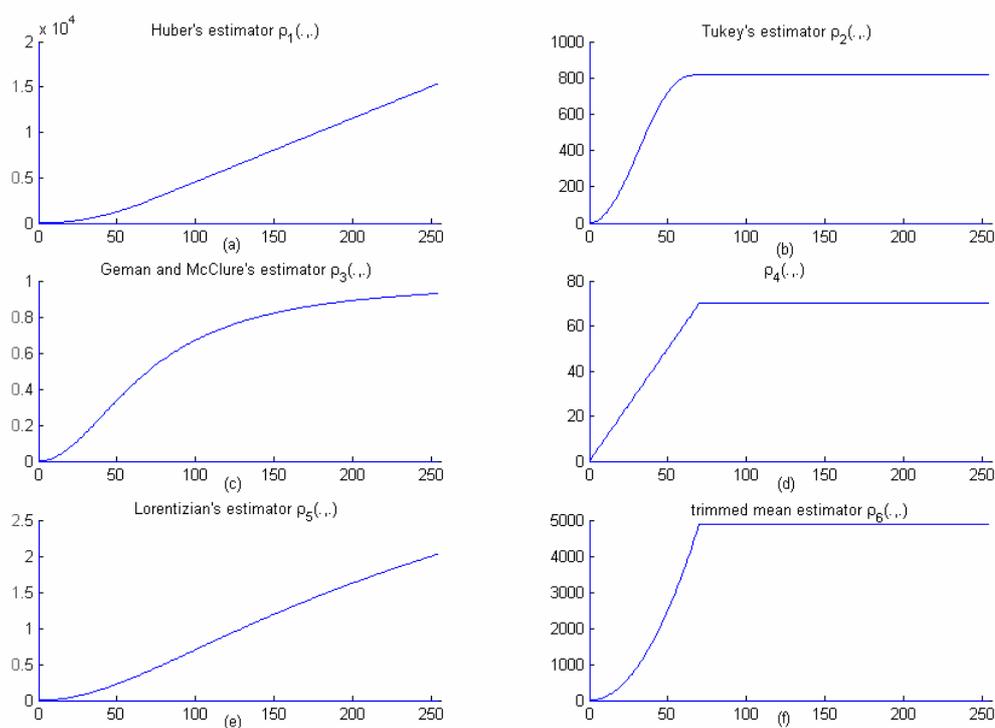


Figure 2. Shapes of commonly used M-estimators with threshold $\sigma=70$. (a) The shape of ρ_1 . (b) The shape of ρ_2 . (c) The shape of ρ_3 . (d) The shape of ρ_4 . (e) The shape of ρ_5 . (f) The shape of ρ_6 .

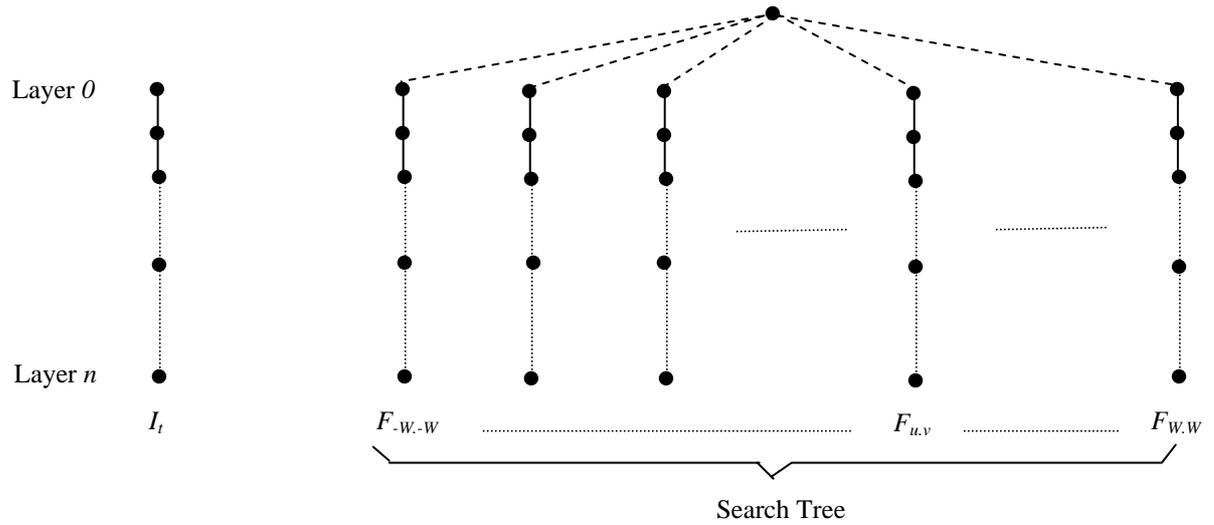


Figure 3. Illustration of the search strategies introduced in Section 3.A. The Li and Salari method [26] only searches the layer 0 and layer n of the tree in a depth-first order. The Lee and Chen method [25] searches the entire tree in a depth-first order. Both methods prune the search branches by comparing the current reference value with the error associated with the vertex. The Chen et al. method [7] uses the uniform cost search [34] (the branch-and-bound strategy) for the entire tree to prune the unnecessary search branches.

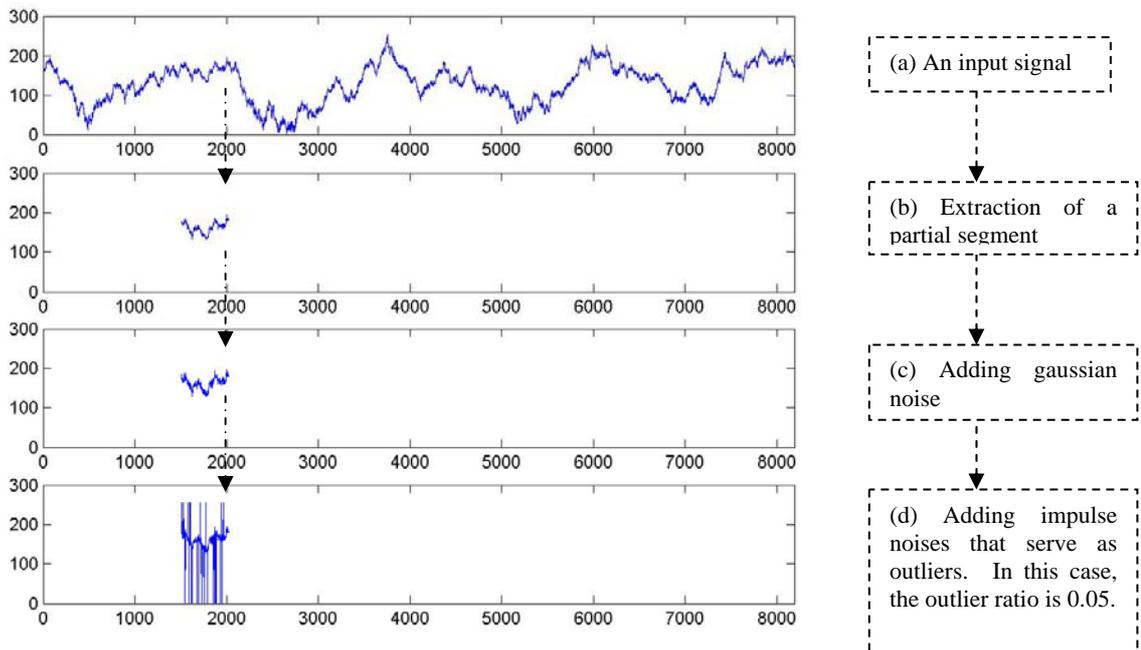
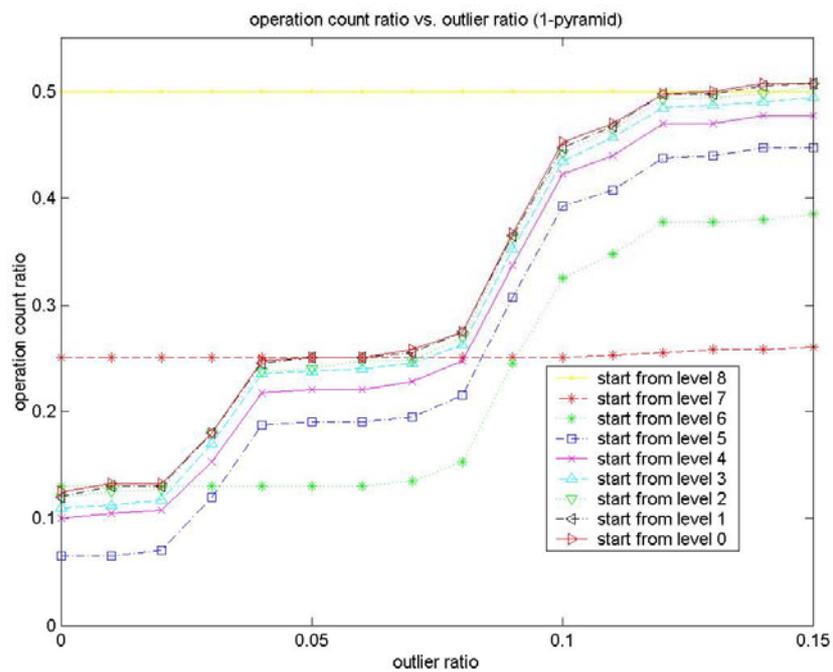
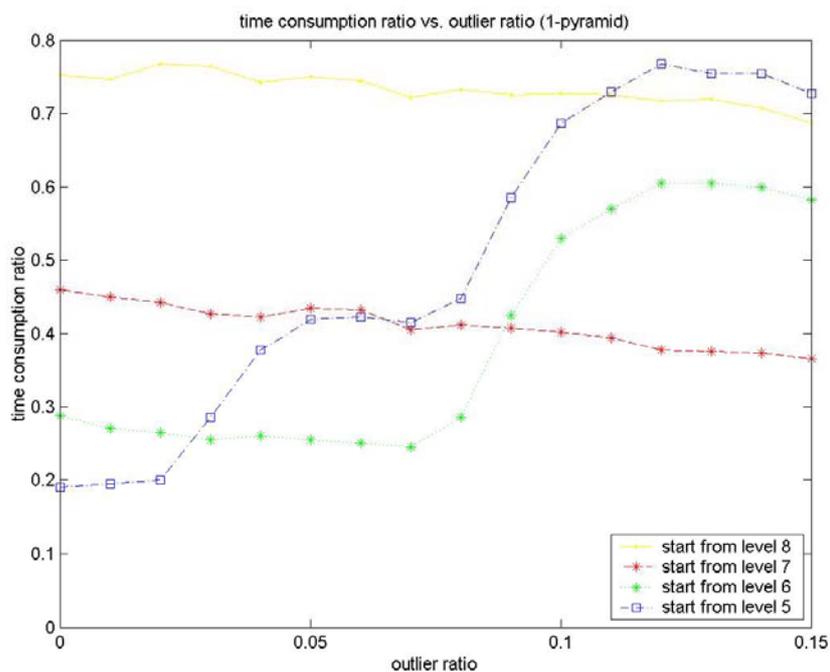


Figure 4. (a) One of the synthetic input signals. (b),(c) and (d) The process of generating a test signal from an input signal.



(a)



(b)

Figure 5. Comparisons between our method and the FS method for robust template matching in the signal matching experiment are shown. Note that simple truncation and the 1-pyramid are used in this experiment. (a) The operation count ratio vs. outlier ratio. (b) The time consumption ratio vs. outlier ratio.

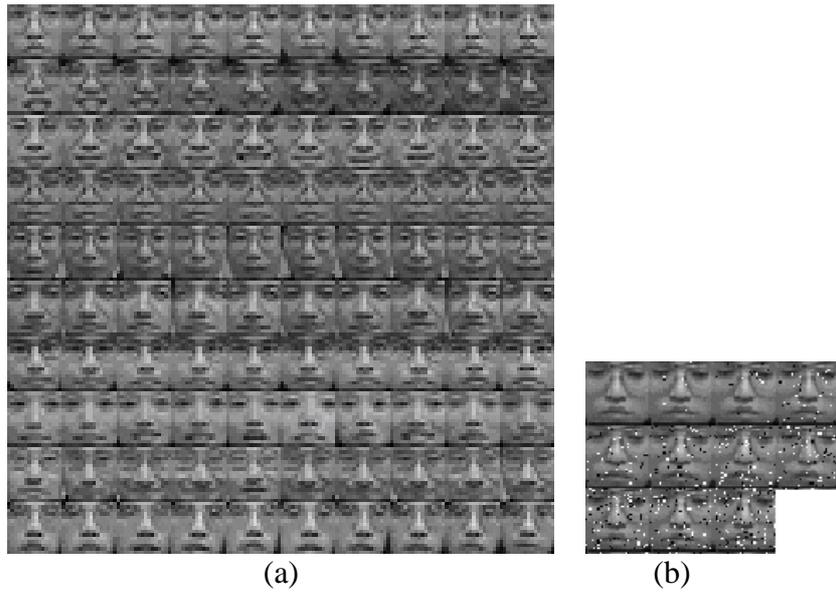


Figure 6. (a) Part of a face-only database used in this paper, showing 100 images from 10 people with 10 images for each person (b) Contaminated images of a person with different outlier ratios. From left to right, top to bottom, the outlier ratios are set from 0 to 0.1.

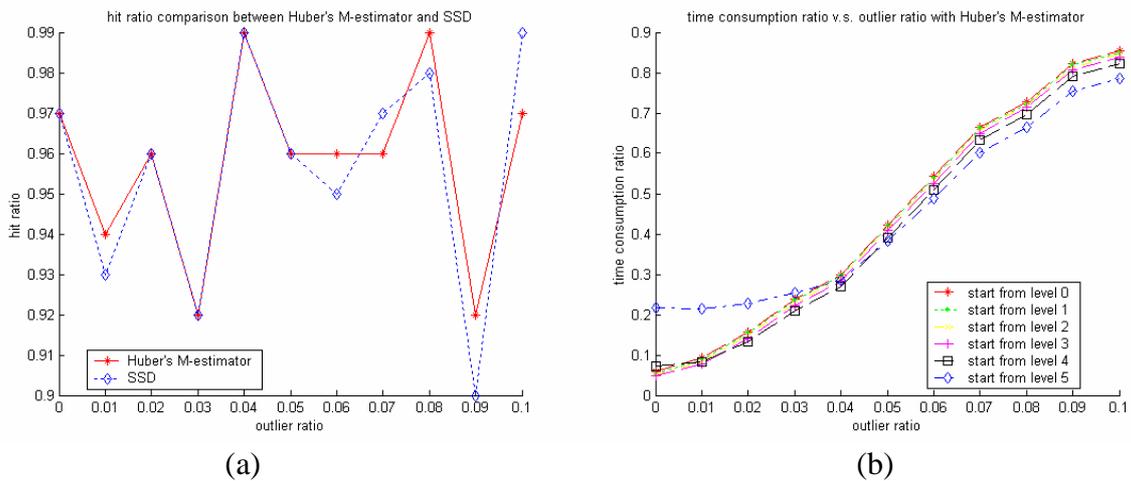


Figure 7. Comparisons between the SSD and SRD using Huber's estimator. (a) The hit ratio vs. outlier ratio. (b) The time consumption ratio vs. outlier ratio.

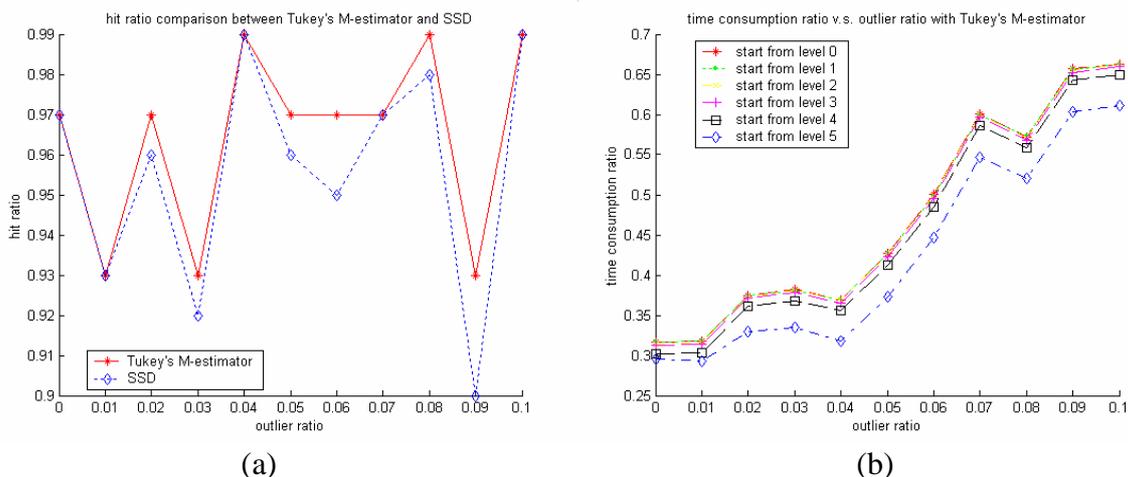


Figure 8. Comparisons between the SSD and the SRD using Tukey's estimator. (a) The hit ratio vs. outlier ratio. (b) The time consumption ratio vs. outlier ratio.

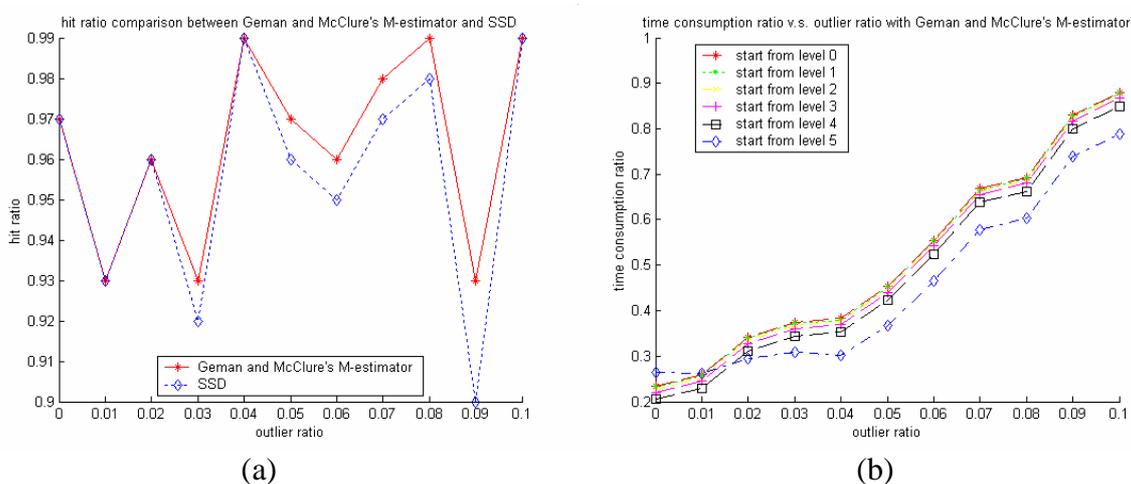


Figure 9. Comparisons between the SSD and the SRD using Geman and McClure's estimator. (a) The hit ratio vs. outlier ratio. (b) The time consumption ratio vs. outlier ratio.

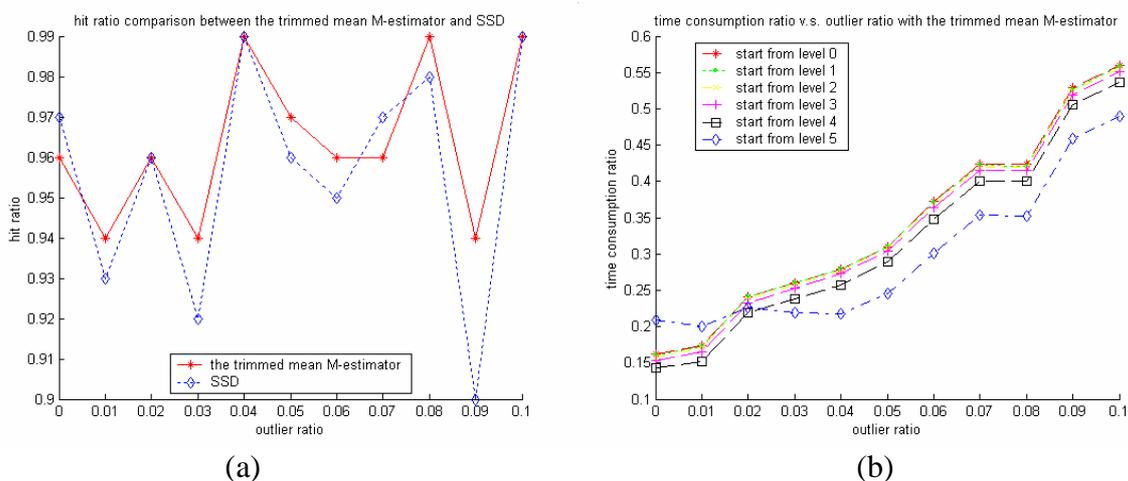


Figure 10. Comparisons between the SSD and the SRD using the trimmed mean M-estimator. (a) The hit ratio vs. outlier ratio. (b) The time consumption ratio vs. outlier ratio.

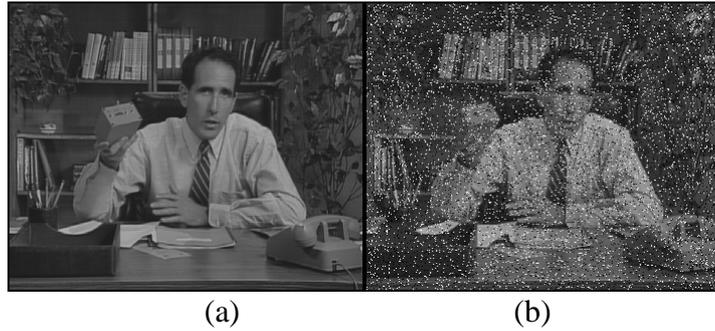


Figure 11. An example of a pair of two consecutive frames for motion estimation. (a) The previous frame. (b) The current frame. Notice that the outlier ratio in (b) is 10%.

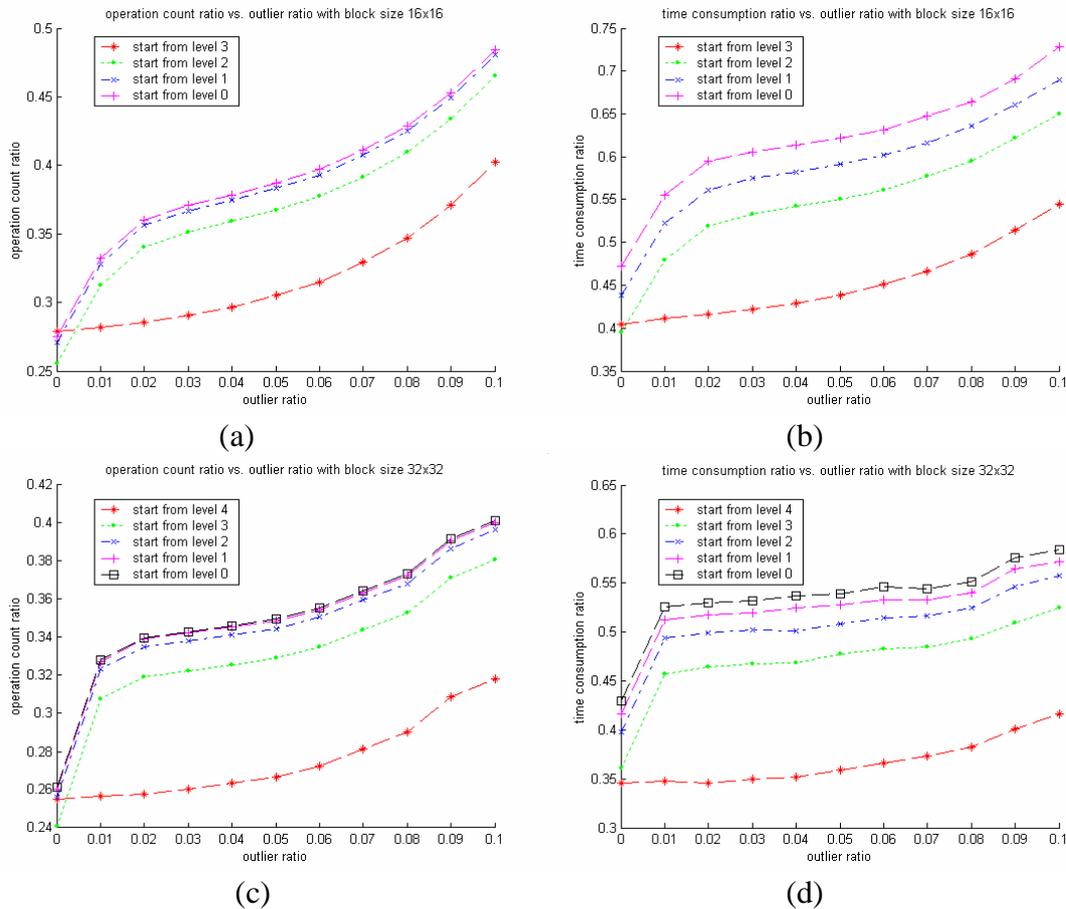


Figure 12. (a) The operation count ratio vs. outlier ratio when the block size is 16×16 . (b) The time consumption ratio vs. outlier ratio when the block size is 16×16 . (c) The operation count ratio vs. outlier ratio when the block size is 32×32 . (d) The time consumption ratio vs. outlier ratio when the block size is 32×32 .