Efficient line voting for MEMS profile registration

Samia BOUCHAFA and Bertrand ZAVIDOVIQUE
Institut d'Electronique Fondamentale
University Paris XI
91405 Orsay Cedex. France

samia.bouchafa@ief.u-psud.fr

Abstract

A new level-line registration technique is proposed for image alignment. This approach is robust towards contrast changes, does not require any estimate of the unknown shift between images and tackles some specific problems due to image acquisition systems: fixed artefacts apparition or repetitive patterns that could lead to paring ambiguities. The registration by itself is performed by an efficient level-line matching process based on a multi-stage primitive election procedure. We deal in this paper with a very challenging situation, due to the particular nature of images: Micro Electromechanical Systems (MEMS) profiles alignement.

1 Introduction

Interferometric profilometry is a microscopy technique which allows to measure the three-dimensional profile of a surface. Resolution is sub-micronic horizontally and nanometric along the vertical axis, likely to further involve sub-pixel refinements. The technique is used for the characterization of $\mu-$mechanical-device behaviours in order to size them up and predict their reliability as electromechanical $\mu-$systems. To extend the field of measurement - limited to a few hundred microns - it becomes necessary to move mechanically the $\mu-$device in front of the lighting and optical system. Then, several partially superimposed 3D profiles are acquired, needing realignment. Due to their 3D nature, the profiles can be represented as pictures and paired thanks to image registration techniques. Aligning profiles then amount to a matching process. However, the specificity of our application, due in particular to various imperfections - dust, spots, stains on lenses - of the measurement system, and varied lighting along with the physical shift, makes it difficult to solve using traditional registration techniques: images do not keep contrast, fixed artefacts do appear and repetitive structures common in such MEMS generate ambiguities against mapping. In addition to this, the implemented method should be fully automated for fundamental-physics users not requiring any arbitrary threshold or a priori knowledge (eg. minimal overlapping, magnitude or direction of deformation).

Automatic approaches for image registration could be classified in different ways depending on the chosen point of view. A very complete classification which takes into account 9 different *criteria* can be found in [6]. However, the most frequent ones use only two categories. The first one spans frequency domain methods and the second one spatial domain ones. Frequency approaches are based on phase correlation and exploit basic properties of the Fourier transform [8]. It is well-known that such techniques are sensitive to brightness changes and require overlapping higher than 50% between registred images. Spatial approaches are based on explicit primitive match (eg. exhaustive search, dynamic warping, etc.), or implicit one (eg. quadratic error minimization). These primitives are points, edges or regions, survey can be found in [1] or [9].

We discard all classical approaches for three main reasons. First, because of their sensitivity to contrast changes. In fact, used primitives either depend directly on images gray-level values (points, regions) or suffer lack of real contrast independant extraction (edges, contours) [3]. Second, most approaches require some initial estimate of the image shift, close enough to the actual solution. Finally, the problem of patterns periodicity is seldom tackled, although resulting into ambiguous matching.

Our approach needs to specifically address such problems. We thus choose to use specific primitives, that is "level lines" of the images. They are robust to contrast changes [3], [7]. They have been used successfully for motion detection in particular with exterior images sequences where the hypothesis of brightness invariance would fail in [2], [5]. In addition, their polygonal nature supports an efficient matching process based on a voting procedure. It consists of constructing a list of potential candidates for each straight piece of level-line, the most similar ones according to an adapted metric and based on specific measurements.

The winning shift cumulates the maximum votes.

Our paper is organized as follows: first we describe our primitive extraction process. Second, we propose a feature selection approach that permits to classify primitives according to their importance. Finally, we present the voting process that allows to exhibit the unknown shift between images. Conclusion outlines satisfactory results we got even in very challenging cases.

2 Level-lines extraction

Let $I(\mathbf{p})$ denote the intensity of the image at the pixel location $\mathbf{p}(x, y)$. The level set λ of I is the set of pixels $\mathbf{p}$ with an intensity equal or greater than λ.

$$\aleph_{\lambda}^{I} = \{\mathbf{p} \diagup I(\mathbf{p}) \geq \lambda \}$$

The level line associated to the level set $\aleph_{\lambda}^{I}$ is its border. One could extract all level sets from the image by using a series of thresholds. Our approach is less time-consuming and more effective: we extract groups of level-lines belonging to the same range of level sets. In fact, we exploit an elementary property of level sets: they are included one in another. That is: $\forall \lambda > \mu \;\; \aleph_{\lambda}^{I} \subset \aleph_{\mu}^{I}$. It means that level lines could be locally juxtaposed but they could never cross. Therefore, we propose a simple recursive extraction process that tracks groups of level lines (the superimposed ones) until they separate. Along the search, subparts are isolated where lines are shown to be straight. This process never considers gray level values direclty but only the relative order between them. It starts at each point $\mathbf{p}_o(x, y)$, determines which ones among its 4 neighbours $\mathbf{p}_o(x, y+1)$, $\mathbf{p}_o(x-1, y)$, $\mathbf{p}_o(x, y-1)$ or $\mathbf{p}_o(x+1, y)$ are successors, initiating four possible paths (see figure 1). Each selected one becomes the current point $\mathbf{p}_k$ and the process repeats until stopping criteria get false.

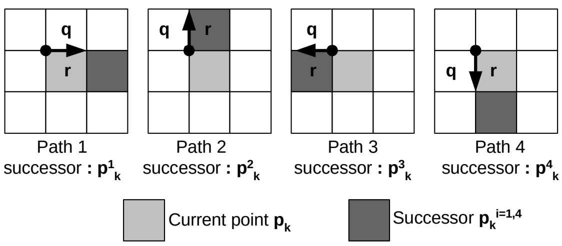


Figure 1. The 4 possible paths starting by a point $\mathbf{p}$, at step k of the recursion, and its 4 possible successors associated with each path. $\mathbf{q}$ and $\mathbf{r}$ represent the points on both sides of the chosen path.

$\mathbf{p}_{i=1,4}^{k}$ is a successor of $\mathbf{p}_k$, at step k if all of the following conditions are verified:

Condition 1 *At least one level line passes between* $\boldsymbol{q}$ *and* $\boldsymbol{r}$*. That is:* $|I(\boldsymbol{r}_k) - I(\boldsymbol{q}_k)| \geq threshold$*. In order to take quantification effects into account , the threshold is set equal to 2 (quantification step + 1). It means that we extract all level lines without any selection at this stage. Let us underline that our primitive extraction process could need no threshold.*

Condition 2 *The tracked level line associated to the chosen path belongs to the same group of level lines being tracked from the beginning (see figure 2):* $[\min(I(\boldsymbol{q}_{k-1}), I(\boldsymbol{r}_{k-1})), \max(I(\boldsymbol{q}_{k-1}), I(\boldsymbol{r}_{k-1}))] \cap [\min(I(\boldsymbol{q}_{k}), I(\boldsymbol{r}_{k})), \max(I(\boldsymbol{q}_{k}), I(\boldsymbol{r}_{k}))] \neq \emptyset$

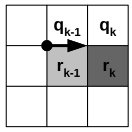


Figure 2. We must be sure that we are tracking the same group of lines that correspond to the same range of level sets.

Condition 3 *The interior (vs. exterior) of the corresponding level sets* $-$ *associated to the tracked level lines* $-$ *is kept on the same side. That is:* $I(\boldsymbol{r}_k)$ *is always greater or always smaller than* $I(\boldsymbol{q}_k)$ *for the same group of tracked level lines.*

Condition 4 *The tracked level lines are still straight. This condition is checked in computing the maximum distance from already traversed points to the cord* [*initial point , current one*]*. This distance should not exceed a threshold (see figure 3). Of course, the already traversed points are not stored. Only the crack code corresponding to the path is saved at each step, the reference system origin being fixed at the initial point* $\mathbf{p}_o$ *in order to calculate the line equation.*

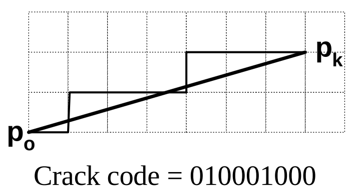


Figure 3. In order to determine if the tracked level lines fit a line, we just compute the distance $d_j = |ax_j + by_j + g|$ between each traversed point $\mathbf{p}_j(x_j, y_j)$ and the line $\overrightarrow{\mathbf{p}_o\mathbf{p}_k}$ of equation $ax + by + g = 0$ where $a^2 + b^2 = 1$.

This iterative process stops when one of the above conditions is not verified any more. Then it is possible to associate to the starting point $\mathbf{p}_o$:

1. the ending point $\mathbf{p}_{end}$
2. the approximated direction of the straight tracked group of level lines $\theta_{\overrightarrow{\mathbf{p}_o\mathbf{p}_k}}$
3. the mean contrast along the tracked level lines: $c = \sum |I(\mathbf{r}_k) - I(\mathbf{q}_k)|$
4. and the length of the tracked level lines: $l = ||\overrightarrow{\mathbf{p}_o\mathbf{p}_k}||$

Using these informations, it is easy to group all points belonging to the same lines. Results are reorganized no longer in terms of starting points but in terms of lines and their characteristics. The end result of the level line extraction process is then the set of all straight pieces of level-lines $\{L_I^i\}_{i=1,N}$ belonging to image I with associated features:

$$\overrightarrow{L_I^i} = [\ \mathbf{p} \quad \theta \quad l \quad c\ ]^T$$

where $\mathbf{p}$=$\mathbf{p}_o$ is one line extremity, θ the orientation, l the length and c the mean contrast along the line.

3 Level-lines Selection

The level-line extraction procedure exhibits all level lines in groups of straight segments belonging to the same range of level sets. However, level-lines do not all have the same importance. Some do separate level sets very close in terms of gray value. This is analog to very low edge slopes when the contrast between regions is light. Other lines are very short, meaning that they are associated to very small level sets likely to be noise generated. Therefore, in the matching process, level-lines are balanced with different weights according to their contrast and length. We choose here to classify the obtained features into 3 categories (from max-priority to min-priority, see figure 4). Let $\bar{l}$,σ_l and $\bar{c}$,σ_c be the mean and standard deviation respectively of all level-line lengths and all level-line mean contrasts over the image I. The first category – most important level lines – correspond to those with high contrast and significant length compared to the means. Let assign them weight $w = W_{\max}$, (in the present application, $W_{\max} = 3$). The weight $w = 2$ is assigned to those having large length too but lower contrast and the last one ($w = 1$) gathers the shortest yet with high contrast. The remaining level lines are rejected (see figure 5). Thus, we can add another characteristic to each line, its priority weight:

$$\overrightarrow{L_I^i} = [\ \mathbf{p} \quad \theta \quad l \quad c \quad w\ ]^T$$

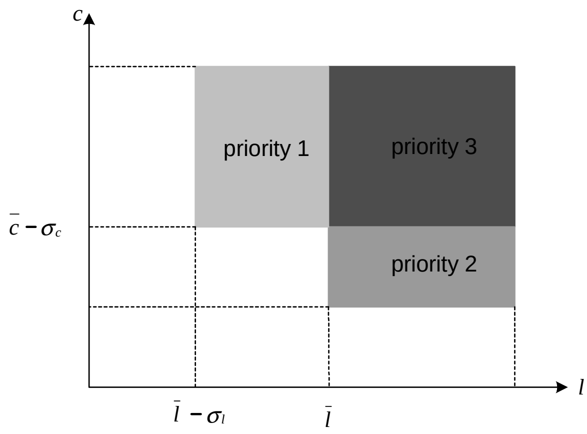


Figure 4. Primitive classification according to their length and their mean contrast from the first category - high priority - (length and contrast above mean contrast and mean length) to the third.

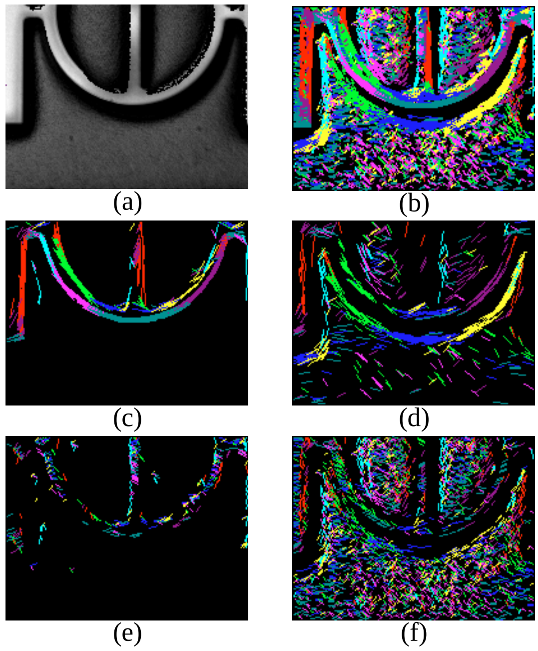


Figure 5. Results of level line extraction and classification. 8 colours are used. For display purpose, each color corresponds to a particular level line orientations using a discretization degree of $\pi/4$. Level lines are displayed roughly using starting point, ending point and obtained direction (a) original image. (b) All straight pieces of extracted level lines. (c) Level line with priority $w = 3$. (d) Level lines with priority $w = 2$. (e) Level lines of priority $w = 1$. (f) Rejected level lines.

4 Level-lines matching

The above level-line classification led to consider a multi-stage matching process where each phase invites a new category of primitives to vote. Before any election, each level-line primitive is asked to construct a $\boldsymbol{\kappa}$-nearest primitive preference list. The voting process then consists of selecting a potential match according to the candidate position in this list. Let us further detail this procedure.

4.1 Preference lists

Each primitive $\mathbf{u} \in \left\{\overrightarrow{L_I^i}\right\}_{i=1,N}$ (straight piece of level line) in the first image I builds a preference list of primitives sorted among those in image J, from most to least similar one, using a distance measure based on length, orientation and average contrast. The distance $dist$ between $\mathbf{u}$ and any $\mathbf{v} \in \left\{\overrightarrow{L_J^i}\right\}_{i=1,M}$ is simply defined as:

$$dist(\mathbf{u}, \mathbf{v}) = k_1 \left|\theta_{\mathbf{u}} - \theta_{\mathbf{v}}\right| + k_2 \left|l_{\mathbf{u}} - l_{\mathbf{v}}\right| + k_3 \left|c_{\mathbf{u}} - c_{\mathbf{v}}\right|.$$

k_1, k_2 and k_3 allow to adjust the relative importance of each characteristic and: $\mathbf{x} = \left[\begin{array}{ccccc} \mathbf{p}_{\mathbf{x}} & \theta_{\mathbf{x}} & l_{\mathbf{x}} & c_{\mathbf{x}} & w_{\mathbf{x}} \end{array} \right]^T$. The size of the preference list is set to $\boldsymbol{\kappa}$. Let $\Im_{\mathbf{u}}$ be the preference list of $\mathbf{u}$. Let $\Im_{\mathbf{u}}[pos] = \mathbf{v}_{pos}$ the primitive of image J at position $pos \in [1, \boldsymbol{\kappa}]$ in the preference list of $\mathbf{u}$ defined by:

$$\Im_{\mathrm{u}}[pos] = \left\{ \begin{array}{c} \mathbf{v}_{pos} \in \left\{\overrightarrow{L_J^i}\right\}_{i=1,M} \diagup \mathbf{v}_{pos} \in \\ \boldsymbol{\kappa}\text{-} \min\left(dist(\mathbf{u}, \mathbf{v})\right) \end{array} \right\}$$

Where the function $\boldsymbol{\kappa}$-$\min()$ returns the $\boldsymbol{\kappa}$ nearest primitives using the above defined distance.

4.2 Voting process

Accounting for periodic patterns in the images requires a multi-stage voting process. Each stage asks a new category of primitives to express their opinion after reducing the number of candidates, until just one shift collects the maximum votes. First, all primitives with high priority weight ($w = 3$) vote. If a single pic appears in the voting space (a shift obtains absolute majority), then the voting process stop. Else a second stage could begin involving an additional population of voters: the primitives with less priority weight ($w = 2$). S pics in the voting space are selected according to the configuration of the space (interdistances, heights, extension...). Voters in this second stage (primitives with weight 3 and 2) have to chose between the selected *maxima*. The voting process stops when no more primitive is avalaible or when just one pic does appear collecting absolute majority.

The voting space V is a cummulative space where coordinates (x, y) represent shift values and the content $\mathrm{V}(x, y)$, or V[$\mathbf{shift}$], is the global vote in favour of shift (x, y). The origin that is identity (**shift** $= 0, 0$) is placed in the middle of the space. The algorithm below describes the main operations of our approach.

{selected shifts} = all the possible shifts
FOR $t = W_{\max}$ DOWNTO 1 DO
    $\forall \mathbf{shift}$, $\mathrm{V}_t\left[\mathbf{shift}\right] = 0$
    FOR each line $\mathbf{u} \in \left\{\overrightarrow{L_I^i}\right\}_{i=1,N}$ in the image I DO
        FOR $pos = 1, \boldsymbol{\kappa}$ DO
            Let $\mathbf{v}_{pos} = \Im_{\mathbf{u}}[pos]$
            $\mathbf{shift} = \mathbf{p}_{\mathbf{v}} - \mathbf{p}_{\mathbf{u}}$
            IF (shift $\in$ {selected shifts}) THEN
                $\mathrm{V}_t\left[\mathbf{shift}\right] = \mathrm{V}_t\left[\mathbf{shift}\right] + \triangle v$
            END
        END
    END
    {Selected shifts} = **S**-$\max\left(\mathrm{V}\left[s\right]_t\right)$
    Stop if one just pic appear in the voting space.
END

The contribution value $\triangle v$ in a vote for a given $\mathbf{shift} = \mathbf{p}_{\mathbf{v}} - \mathbf{p}_{\mathbf{u}}$ between two primitives $\mathbf{u}$ and $\mathbf{v}$ is computed through four parameters. Let $\triangle v = \alpha \times \beta \times \gamma \times \delta$, parameters α, β, γ and δ are ajusted as follows :

1. A potential correspondent $\mathbf{v}_{pos}$ will have a higher contribution if it lays in the begining of the preference list. α is thus inversely proportional to the position pos of the primitive $\mathbf{v}_{pos}$. Let choose:

$$\alpha = \frac{\boldsymbol{\kappa}}{pos} \quad \alpha = \{1, ..., \boldsymbol{\kappa}\}$$

2. At a given stage of the voting process, primitives with priority greater or equal to $t, t \in \left[W_{\max}, W_{\min}\right]$, are allowed to vote. For example, in the first pass t= **3** and voters must have priority 3 ; for the second pass t=2 and voters must have priority 2 or 3, etc**.** However, primitives with higher priority still vote in higher consideration. β is then proportionnal to the primitive weight. That is:

$$\beta = \frac{w_{\mathbf{u}}}{W_{\max}} \quad \beta = \left\{\frac{1}{W_{\max}}, ..., 1\right\}$$

3. In most cases, the number of primitives with higher priority (reliable primitives) is smaller than other categories. Thus, we have to normalize the vote of a primitive belonging to a given category with respect to the

number of reliable primitives ($w_{\mathbf{u}} = W_{\max}$), in order to reduce its contribution. Let γ be the normalization factor.

$$\gamma = \frac{P(w_{\mathbf{u}_i} = W_{\max})_{i=1,N}}{P(w_{\mathbf{u}_i} = w_{\mathbf{u}})_{i=1,N}} \quad \gamma \in]0...1]$$

4. Finally, at a given pass t, a shift is given a voting value $\mathrm{V}_t\,[\mathbf{shift}]$ to be taken into account in the next pass. δ is then the relative value of the shift compared to the maximum value obtained in the previous voting stage:

$$\delta = \frac{V\,[shift]_{t-1}}{\max\left(V\,[s]_{t-1}\right)} \quad \delta \in]0...1]$$

5 Results

Following figures give first results of our alignment method in three different cases of images with very large amplitude shift. Second and third couple of images are more challenging due to the repetitive patterns, and low contrast.

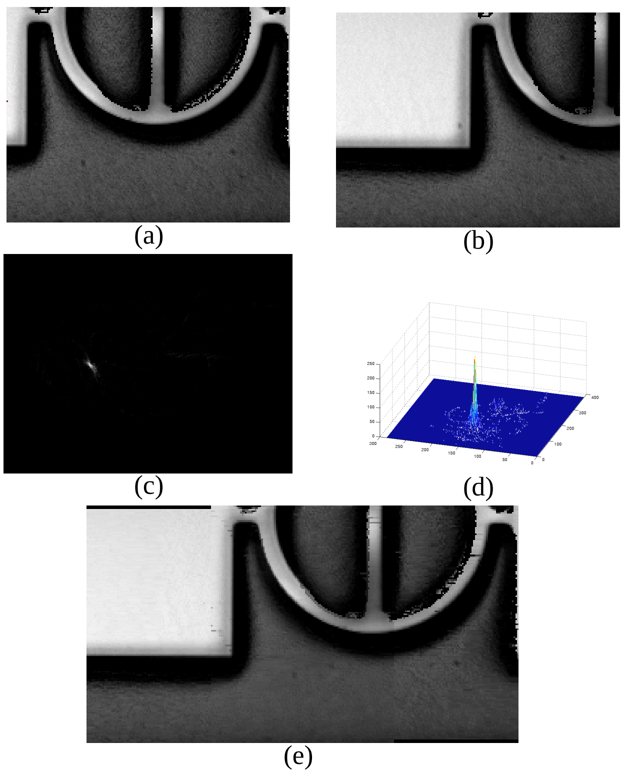
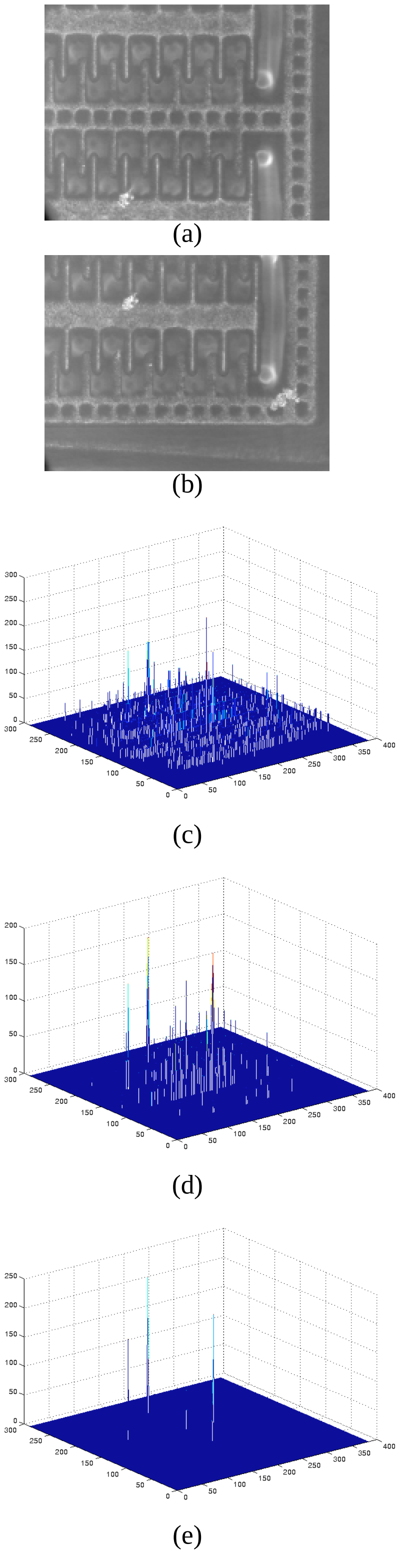
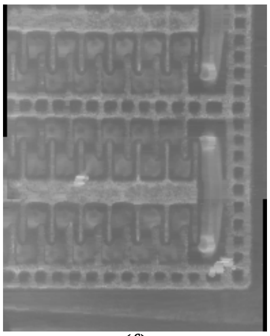

(a) (b) (c) (d) (e)

Figure 5. (a) and (b) Two partial profiles of an aluminium ring. (c) and (d) The voting space represented by an image or a 3D mesh and its maxima which correspond to the different possible shifts. Here, only one pass is used because one significant enough pic appears in the voting cumulative space. (e) the result of our alignment method : shift of (-2,72) which corresponds to the shift pic.


(a) (b) (c) (d) (e)



(f)

Figure 6. (a) and (b) Partial profiles of a micro gyrometer electrostatic copper comb. (c) to (e) The voting spaces corresponding to the 3 vote iterations. (f) The resulting correct alignment (shift of -98, 3) in spite of the repetitive patterns.

6 Comparision with other systems

We have compared our technique with two classical points-based approaches: phase correlation (using an FFT transform) and intensity distance minimisation (using the well-known Levenberg-Marquardt optimisation method). None of them give expected results. They require either a significant covering between images or a good estimate of the solution. We have also used a very interesting online web demo [4]. Due to the repetitive patterns present in the images, the obtained shift is not correct (see Figure 7).

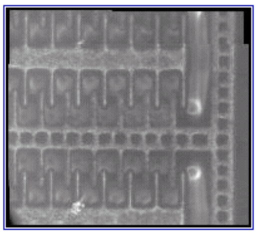


Figure 7. The resulting shift=(28, 1) obtained by the UCSB System [4] using images of Figure 6.

7 Conclusion

We have proposed an image registration approach robust to contrast changes, that does not require any prior estimate of the unknown shift between images and that takes into account specific problems due to our image acquisition system or to the application, like lighting movements and repetitive patterns. We chose specific primitives with a very suitable property: their contrast change invariance. We then propose a recursive extraction process that never uses direclty gray level values but only the relative order between them. The registration by itself is performed through an efficient matching procedure based on multi-pass voting. It consists of constructing a list of potential candidates for each straight piece of level-line, the most similar ones according to an adapted metric and based on specific measurements. The winning shift cumulates the maximum votes. The multi-pass voting is introduced in order to tackle the problem of local maxima − corresponding to the repetitive pattern responses − that could be found in the voting space. Primitives are then invited to vote according to their importance at each pass of the election. Results we got are very encouraging and give expected results in spite of the very challenging situations we have considered. Our next step is to make the process more robust to perturbations, up to adress other transformations between images like rotations or scaling.

Acknowledgement *We are very grateful to Mr Alain Bosseboeuf, CNRS researcher at University Paris XI and his team for providing us with these very suitable data.*

References

[1] L.G. Brown. *A survey of image registration techniques*. ACM computing surveys, 24(4), 325−376, 1992

[2] S. Bouchafa. *Motion detection invariant to contrast changes. Application to detecting abnormal motion in subway corridors*. Phd Thesis. University Paris 6, 1998.

[3] V. Caselles, B. Coll, J. Morel. *Topographic maps & local contrast change in natural images*. IJCV, sept. 1999

[4] F. Dmitry, C. Kenney, B.S. Manjunath, L.M.G. Fonseca. *On line registation demo*. http://nayana.ece.ucsb.edu/registration/, University of California Santa Barbara, 2002.

[5] F. Guichard, S. Bouchafa, D. Aubert. *A change detector based on level sets*. ISMM2000, Palo Alto, june 2000.

[6] J.B.A. Maintz, M.A. Viergever. *A Survey of medical Image Registration*. Medical Image Analysis, Oxford University Press, Volume 2, 1998.

[7] P. Monasse, F. Guichard. *Fast computation of a contrast-invariant image representation*. IEEE Trans. on Image Proc., 9(5), 860−872.

[8] B.S. Reddy, B.N. Chatterji. *An FFT-based technique for translation, rotation and scale-invariant image registration*. IEEE Trans. on Image Proc., 5(8), 1996.

[9] R. Szeliski. *Image mosaicing for tele-reality applications*. In Proc. of IEEE Workshop on applications of computer vision, 1994