

6-17-2013

Correlation of partial frames in video matching

Boris Kovalerchuk

Sergei Kovalerchuk

Follow this and additional works at: <https://digitalcommons.cwu.edu/compsci>



Part of the [Graphics and Human Computer Interfaces Commons](#)

Correlation of partial frames in video matching

Boris Kovalerchuk^{ab1}, Sergei Kovalerchuk^b

^a Dept. of Computer Science, Central Washington University, Ellensburg, WA 98926-7520,

^b BKF Systems, 8241 S. 123 St., Seattle, WA, 98178

ABSTRACT

Correlating and fusing video frames from distributed and moving sensors is important area of video matching. It is especially difficult for frames with objects at long distances that are visible as single pixels where the algorithms cannot exploit the structure of each object. The proposed algorithm correlates partial frames with such small objects using the algebraic structural approach that exploits structural relations between objects including ratios of areas. The algorithm is fully affine invariant, which includes any rotation, shift, and scaling.

Keywords: Image processing, video processing, partial frames, correlation, video matching, affine invariance, sensor fusion.

1. INTRODUCTION

The correlation of the data from multiple EO/IR and radar sensors on different platforms and at different locations is critical for the success of data fusion tasks. In general, the data fusion tasks deal with heterogeneous information, disparate data sources that may contain correlated, shared and/or contradictory information. These tasks are known at different levels from the signal level to the decision making level [Liggins et al, 2008].

This paper is focusing on the correlation of both partial and complete EO/IR frames. The objects on these frames are at large distances from the sensors, including a very difficult case, where each object's size is only a single pixel. In this work, a frame is considered a partial frame if a part of the frame lost any information, a frame is shifted, rotated, or sheared relative to the line of sight (LOS) of the previous frame, which is under an unknown affine transform. In addition to that, the two frames can be corrupted by noise and corrupted differently. We focus on small objects (1-2 pixels each) with about 10 objects in the frame and the angle between the lines of sights of the sensors being within 90° .

Incorrect correlation of multiple EO/IR sensors can distort a conclusion about the capabilities of sensor systems. The presence of "image chips" or partial frames is an important source of incorrect correlation of sensor messages. Partial frame correlation also opens an opportunity for decreasing the image data transmission load. It can be done by transmitting only a compressed partial frame or even as a vector file instead of a much larger raster file of the full frame. Downloading partial frames requires less bandwidth, but their correlation is more difficult especially for frames with objects at long distances that are visible as single pixels where the algorithms cannot exploit the structure of each object. Frame correlation is a part of the *video matching* area [Lee et al, 2009] that is growing to meet the needs in:

- (1) *video retrieval* (to retrieve video clips, similar to a query clip [Li et al, 2006; Yuan et al, 2007],
- (2) *matching frames* from multiple cameras/sensors in time and space [Bar-Shalom, Blair, 2000; Liggins, et al, 2008]), and
- (3) *watermarking* (to synchronize original and watermarked videos before the extraction of frame-dependent watermarks [Delannay et al, 2004]). The required accuracy of video matching is lower for video retrieval than for watermarking, because some systems use the pattern of skipped frames as a watermark [Lee et al, 2009].

¹borisk@cwu.edu; phone/fax 509 963-1438

The objective of this paper is creating *the rigorous models and software* to investigate and mitigate *the* impacts of correlating EO/IR sensor messages under non-ideal conditions of Partial Frames (COPAF). These models and software must be of high fidelity and of high computational efficiency and scalability based on the use of modern advanced algorithms, efficient multi-threading, and modern multi-core CPUs and GPUs. The proposed rigorous solution is based on an **affine invariant area ratio method** to these requirements. Partial frame correlation is part of a more general task of *correlating data from multiple sensors* (for short, *sensor correlation (SC)* task).

There are several ways to combine the Sensor Correlation and the Object Identification tasks. One of them is matching objects from multiple sensors using a sensor correlation method at the first step, and then identifying already matched object to categories. An alternative way is identifying objects from each sensor first, and then matching identifying objects among different sensors. The system architecture, bandwidth and other factors impact the choice one of these options. In both cases it is critical to have a robust sensor correlation method.

The related work includes studies in areas know as affine invariant image matching, correlation, co-registration, and conflation [Flusser, Suk,, 1994; Zitova, Flusser, 2003; Morel, Yu, 2009; Yu, Morel. 2011; Micolajczyk et al, 2005; Paradowski, Śluzek, 2010, 2011; Pang et al, 2012; Liu et al, 2012; Kang et al, 2006; Lee et al, 2009, Kovalerchuk et al, 2005, 2007, 2009, 2011, Delannay ety al, 2003]. In [Kang et al, 2006] triangle invariants have been used for automatic SAR image registration. The COPAF method also used the invariant based on the triangles. The algorithm in [Kang et al, 2006] departed from selecting 3 top matches used in [Flusser, Suk, 1994] for a procedure that involves more information. The COPAF also uses additional information but differently. This algorithm differs also from the algorithm described in [Kang et al, 2006] in the way how triangles are produced and tested influenced by the differences in the tasks. Specifically this work is focused on small objects of pixel size and partial video frames.

2. METHOD

2.1. Structural approach and steps of the algorithm

The COPAF algorithm is applicable for situations with two or more sensors simultaneously (a) observing the overlapping areas with multiple objects or (b) observing a single object in each frame. In the first case, the algorithm deals with blobs extracted in the frame from multiple objects. In the second case, it deals with blobs extracted from a single object.

The main idea of the algorithm is correlating frames using the structural information in these frames using the algebraic structural approach [Kovalerchuk et al, 2005, 2007, 2009] that exploits structural relations between elements of the frames. The structural relations are presented matrixes. The matrix representation is important because it permits us to convert the situation to a generic **algebraic system framework**, with algebraic systems $A_k = \langle A_k, R_k, \Omega_k \rangle$, where signature Ω_k contains the operator $V_k(a_i, a_j)$ represented as a matrix V_k and handles the correlation problem uniformly. From this point, uniformity permits us to use a single algorithm to search for matching features in the images. It does not matter for the algorithms in the algebraic form whether the elements of A_k are straight-line segments, polylines, areas, or complex combinations, or some other features. Elements of A_k also can be numeric characteristics of image components such as the size of region i in image k , S_{ki} that are used in the COPAF algorithm proposed in this paper.

Figure 1 shows the overall upper-level COPAF system design. The *main steps* of the COPAF algorithm to correlate two partial or full frames from two different sensors at time t are:

- (1) extracting at least three prominent areas (blobs) in each frame,
- (2) computing centroids of these blobs (centroids typically are very robust even when the edges of the blobs differ from frame to frame),
- (3) building multiple triangles that use centroids as vertices,
- (4) computing areas of each triangle,
- (5) computing ratios of these areas within each frame,
- (6) using the theorem that these ratios are fully affine invariant to find a matching pair of triangles in another frame (the match is identified by similarity threshold for the ratios and by the test on other blobs),
- (7) computing centroids of matched triangles called t -centroids,
- (8) selecting three t -centroids with the best similarity match,
- (9) building a 2-D affine transform F for the selected t -centroids from the first frame to the second frame,

- (10) transforming the first frame using the transform F ,
- (11) overlaying the transformed frame 1 onto frame 2 to find the overlapping area,
- (12) correlating the frame in the overlapping area pixel by pixel to confirm the transform F .

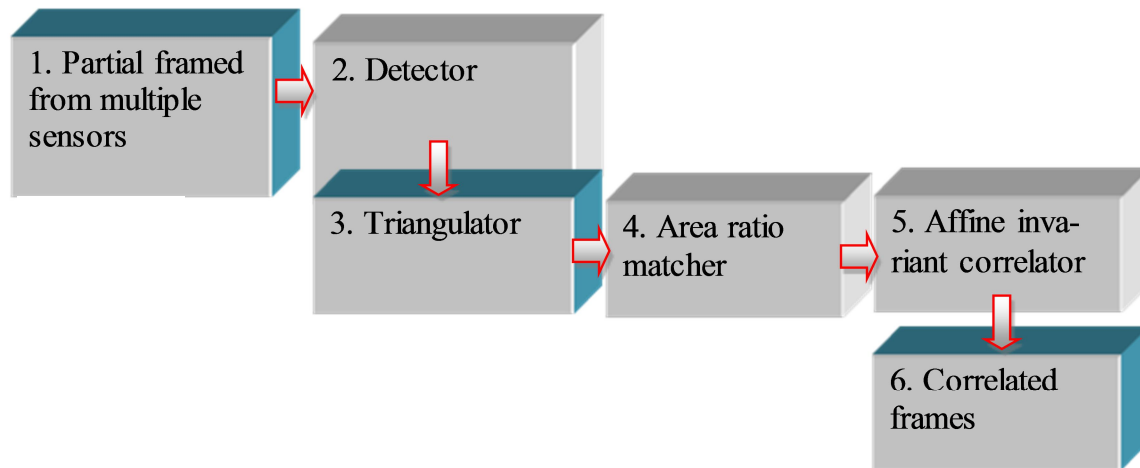


Figure 1. Overall upper-level COPAF system design.

2.2. Area ratio-based affine invariant approach

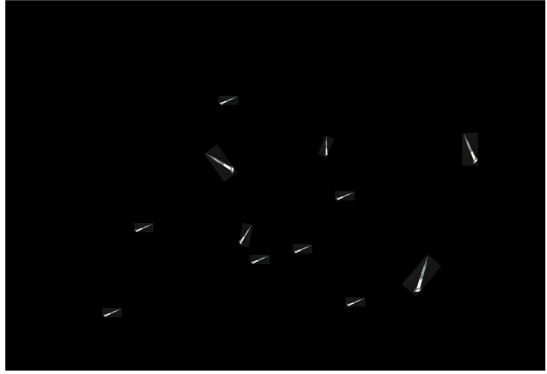
The COPAF algorithm finds matching objects in another frame by using the fundamental theorem that ratios of areas are fully affine invariant [e.g., Flusser, Suk, 1994; Zitova, Flusser, 2003; Hill, Kelley, 2007; Kovalerchuk, Schwing, 2005]. A formal expression of this statement as the *invariance theorem* is presented below. Let G_1 and G_2 be two closed regions and $S_1=S(G_1)$ and $S_2=S(G_2)$ be their areas respectively, where $S(\)$ is an operator that computes the area of the region G_i .

Theorem: An affine transformation F of the image does not change the relation between area ratios, $S_1/S_2 = S(G_1)/S(G_2)=S(F(G_1))/S(F(G_2))$, that is F is an *isomorphism* for the area ratio $S(G_1)/S(G_2)$.

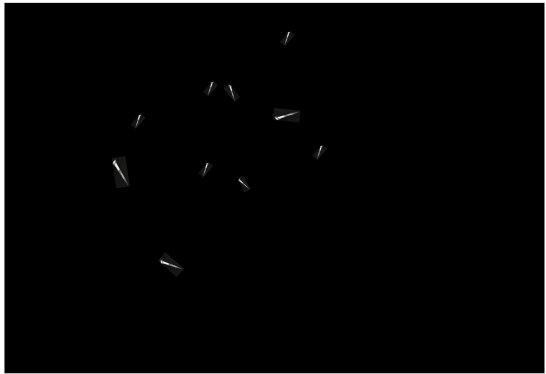
While this mathematical statement is valid, to apply it we need to solve several problems. The first one is to find or generate such regions S with distinctive ratios to be robustly exploited. The second one, noise and other factors will corrupt values of ratios, thus the ideal mathematical equality of ratios will be violated. Therefore, a similarity measure between ratios must be established. Figures 2 and 3 illustrate the algorithm to improve detection, which is described above.

Figure 2 shows how such regions have been generated from the images that have no large regions, but only small individual detections (Figure 2 (a)-(d)). These detections are used to find the centroids of triangles built by triangulation of detections (Figure 2(e)-(f)) by the algorithm that we propose. Figures 2(e), and (f) show only some of generated triangles. These triangles selected by the algorithm in (e) are matched with triangles in (f) by the area ratios. The centroids of matched triangles are shown in (e) and (f) in the same color. Then these centroids are used to build an affine transform F by using classical mathematical techniques. Figure 2(h) shows the result of the application of F to the whole set of points in Figure 2 (f). Next Figure 2(g) shows (c) and (h) superimposed with red points from (c) that are absent in (h). Figure 3 illustrates the application of the algorithm for a single object viewed by two sensors where the matching triangles are built by using centroids of blobs extracted from the object.

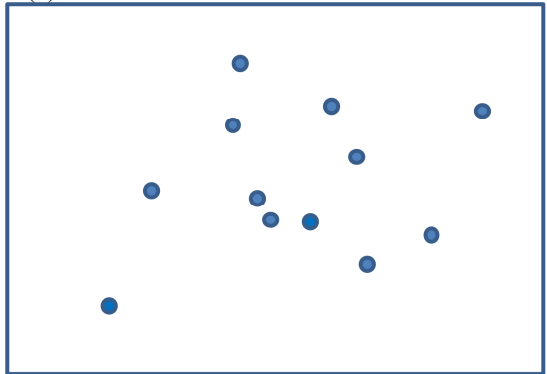
Area Ratio Conflation (ARC) Algorithm [Kovalerchuk, Schwing, 2005]. If two areas in the image A have a ratio, say $1/3$, then the same ratio between them should remain the same under any affine transform of A . Thus in the second image, we can compute areas of regions, their ratios S_i/S_j and search for a $1/3$ ratio among them.



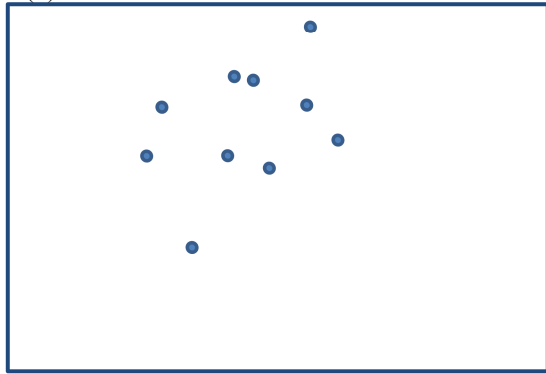
(a)



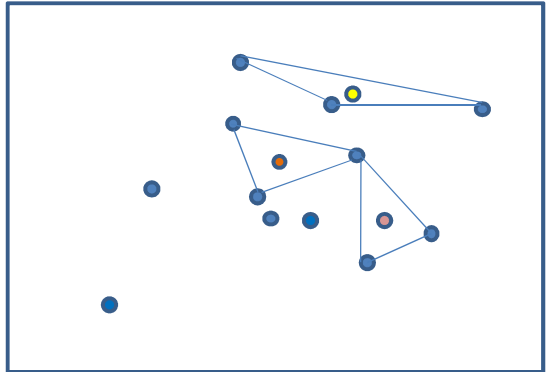
(b)



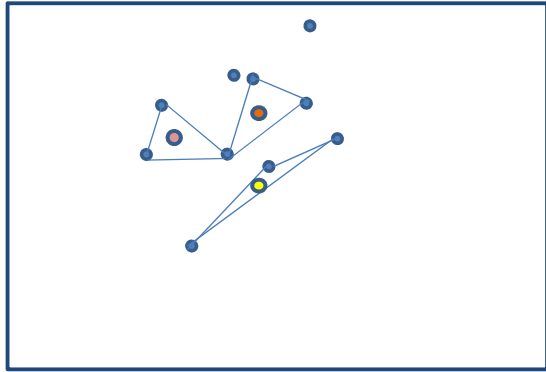
(c)



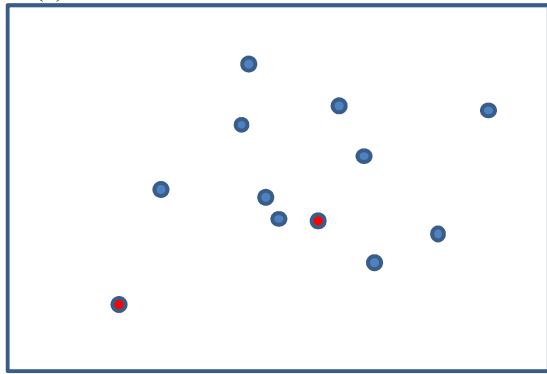
(d)



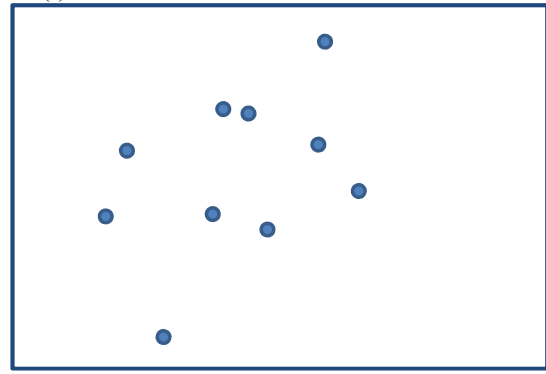
(e)



(f)



(g)



(h)

Figure 2. Multi-object detection by two sensors with discovery of rotation and missing objects (red).

If only one such ratio was found, then the centers of these regions give us two tie (control) points for building an affine transform. Finding a third region S_m in both images with the equal ratios S_i/S_m in both images provides the third tie point needed for an affine transform. This basic idea is adjusted for the cases where more than one matching triple is found. An additional uniqueness criterion is introduced in the algorithm based on the analysis of additional ratios.

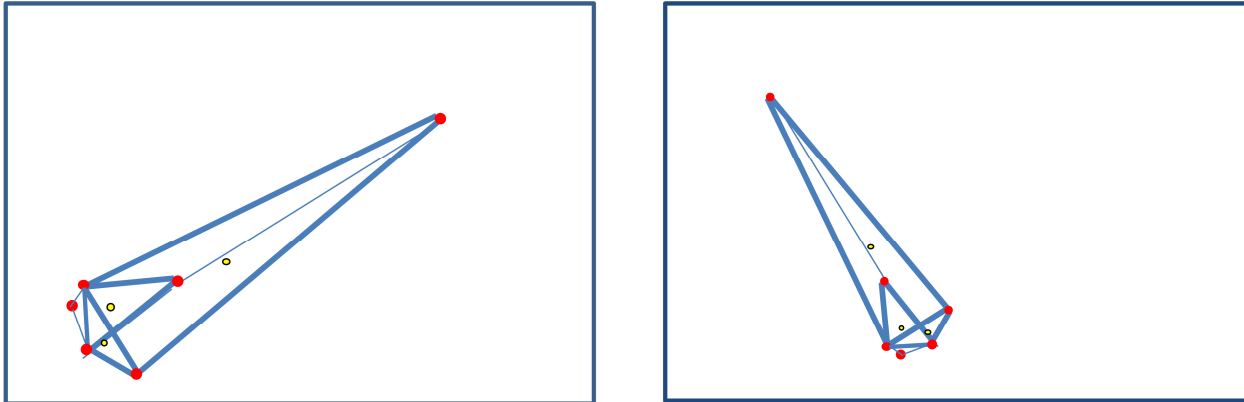


Figure 3. Single object detection by two sensors.

While this size ratio ($1/3$) is invariant to affine transformations, the ratio precision needs to be adjusted to the scale of the least precise image. Ratios $1/3$, $1/2$ and $1/4$ could match 0.336 0.52 0.27 if images are of different scales. The algorithm uses a matching threshold for these cases. The *Ratio algorithm* computes areas for each region in both images, $S_{1i}=S(G_{1i})$, $S_{2i}=S(G_{2i})$ as a number of pixels inside of the region. Next, this algorithm computes two matrixes V_1 and V_2 . Elements of matrix $V_1=\{c_{ij}\}$ are $c_{ij}=S_{1i}/S_{1j}$. Elements of matrix $V_2=\{q_{ij}\}$ are defined similarly, $q_{ij}=S_{2i}/S_{2j}$. We assume that all areas S_{1i} and S_{2i} are positive. As was stated before, the matrix representation is important because it permits us to convert the situation to a generic **algebraic system framework**, with algebraic systems $\mathbf{A}_k=\langle A_k, R_k, \Omega_k \rangle$, where signature Ω_k contains the operator $V_k(a_i, a_j)$ represented as a matrix V_k and handles the correlation problem uniformly for straight-line segments, polylines, areas, or complex combinations, or some other features.

Example: Let matrix V_1 be computed for regions with areas $S_{11}=6$, $S_{12}=4$, $S_{13}=2$, $S_{14}=1$ (see Table 1) in image 1 and matrix V_2 is computed for areas $S_{21}=4$, $S_{22}=1$, $S_{23}=6$, $S_{24}=7$ in image 2 (see Table 2).

Table 1. Matrix of shape size ratios in Image 1.

	$S_{11}=6$	$S_{12}=4$	$S_{13}=2$	$S_{14}=1$
$S_{11}=6$	1	4/6	2/6	1/6
$S_{12}=4$	6/4	1	2/4	1/4
$S_{13}=2$	6/2	4/2	1	1/2
$S_{14}=1$	6/1	4/1	2/1	1

Table 2. Matrix of shape size ratios in Image 2.

	$S_{21}=4$	$S_{22}=1$	$S_{23}=6$	$S_{24}=7$
$S_{21}=4$	1	1/4	6/4	7/4
$S_{22}=1$	4/1	1	6/1	7/1
$S_{23}=6$	4/6	1/6	1	7/6
$S_{24}=7$	4/7	1/7	6/7	1

The brute force method would search equal ratios in two matrixes excluding the diagonal. This is an unambiguous case, where the ratio 6/4 for $S_{11}=6$, $S_{12}=4$ is matched to the same ratio for $S_{23}=6$, $S_{21}=4$, that is the region G_{11} in image 1 is matched to the region G_{23} in image 2, and the region G_{12} in image 1 is matched to the region G_{21} in image 2. The centroid of each region is computed as an average of coordinates of all points (pixels) of the region.

The computational efficiency of the algorithm depends on how quickly matrixes V_1 and V_2 will be computed for images with the large number of regions. It is reasonable to start from $n-1$ independent ratios in the first row. If all these ratios in A differ from $n-1$ independent ratios in B then the next $n-2$ ratios are computed in both matrixes. If they also have no equal values then the process continues for the next $n-3$ ratios until all elements A from the upper part of A are exhausted. If some ratios in this process are equal then there is no reason to compute ratios that are derived from them. They will be equal too. Specifically, if $c_{ij} = q_{st}$ and $c_{jk} = q_{tq}$ then we do not need to compute c_{ik} and q_{sq} . They do not add new information and are equal (see proof below). From $c_{ij} = q_{st}$ we can derive that region G_{1i} is matched to the region G_{2s} , and the region G_{1j} is matched to the region G_{2t} . More details on efficient computations are provided in [Kovalerchuk, Schwing, 2005]. To make the match more visible we can reorder the lines in Tables and 2 for human inspection.

In mathematical terms, we need to find V_{max} a largest common subset in matrixes V_1 and V_2 . If this subset is at least 3×3 , i.e., it includes three features and centers of those features are not located on the same straight line then an affine transform can be found between images 1 and 2. If the common subset is much larger than 3×3 , then it creates a higher level of confidence that the match of two images is not accidental and can be used for testing affine transforms.

An algorithm generates all subsets of V_1 of size 3×3 and all subsets of V_2 of size 3×3 . We denote them as $V_1(T_r)$ and $V_2(T_e)$, where T_r and T_e are index sets that identify three features that form a 3×3 subset. Then the similarity of every pair $V_1(T_r)$ and $V_2(T_e)$ is tested using the Euclidian distance (D):

$$D(V_1(T_r), V_2(T_e)) = \sum_{i,j \in T_r} (c_{ij} - q_{\varphi(i), \varphi(j)})^2$$

and a pair is found with the smallest distance that is below the threshold L , where the mapping φ matches the features in two images:

$$D(V_1(T_r), V_2(T_e)) \rightarrow \min$$

$$D(V_1(T_r), V_2(T_e)) \leq L.$$

After the match and conflation affine transform are done by the ARC algorithm the **conflation/alignment quality** is evaluated by using the difference of two regions (XOR of pixels of regions G and G'). The **absolute difference** of regions G and G' , $\Delta(G, G')$ is computed as the number of pixels in the difference of regions G and G' : $\Delta(G, G') = S(\text{XOR}(F(G), G'))$ and the **relative difference** of the regions is $\rho(G, G') = \Delta(G, G') / (S(G) + S(G'))$. The total difference between three matched regions $\{G\}$ and $\{G'\}$ of $\text{Im}1$ and $\text{Im}2$ found by the ARC algorithm is $\mu(\{G\}, \{G'\}) = \mu(G_1, G'_1) + \mu(G_2, G'_2) + \mu(G_3, G'_3)$. Similarly the relative difference of matched regions is $\rho(\{G\}, \{G'\}) = [\sum_{i=1,2,3} \Delta(G_i, G'_i)] / [\sum_{i=1,2,3} (S(G_i) + S(G'_i))]$. The relative difference is in the $[0, 1]$ interval.

3. ARCHITECTURE FOR CORRELATING PARTIAL FRAME

The architecture depends on the goal of the system to support correlation of frames from a single sensor or from multiple sensors. This leads to: (1) *Separate architecture* that deals with multiple consecutive frames from each *single sensor* and (2) *Integrated architecture* that deals with multiple frames from *multiple sensors* at multiple times.

Architecture for a single sensor. A single sensor architecture alternative is outlined in Figure 4, where a frame F at time t , $F(t)$ is incomplete and frame $F(t+k)$ at time $t+k$ is complete but oriented incorrectly as next frames after it indicate. The COPAF architecture for this task uses frame $F(t+k)$ to complete frame $F(t)$. Then frame $F(t)$ is used to reorient frame $F(t+k)$. Figure 5 illustrates this architecture with the described scenario where frame 1 is partial e, frame 2 is also partial, but without cutting the object, and frame 3 is a complete frame.

The advantage of this single sensor architecture is in the ability to execute frame correlation *in parallel for each sensor* on different computing resources, and even on each sensor platform locally, if the platform is equipped with the required computing resources.

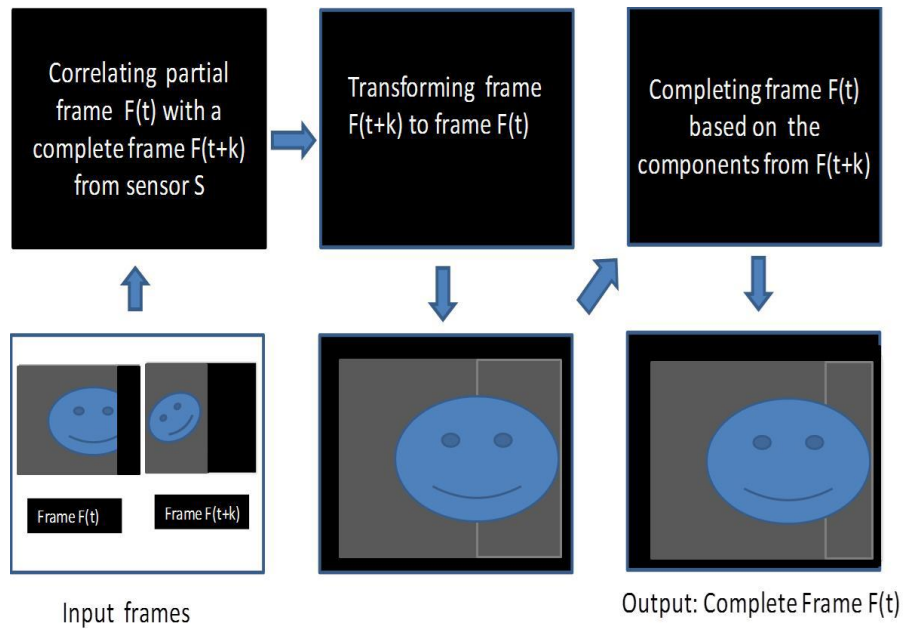


Figure 4. Single sensor architecture.

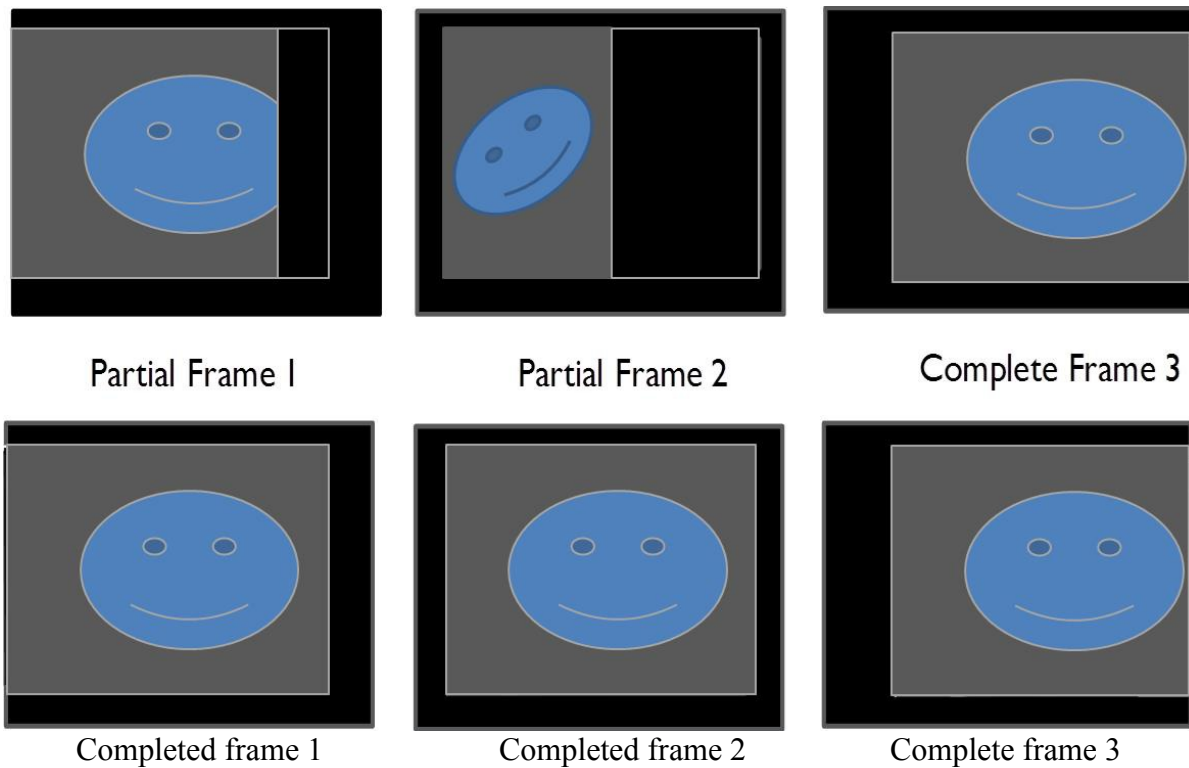


Figure 5. Experiment design for restoration of consecutive frames for a single sensor by correlating partial frames.

Architecture for two or more sensors at a given time. This architecture is to correlate simultaneous partial frames from two or more sensors. The two-sensor architecture for a given time t is shown in Figure 6 for sensors F and Q . This figure illustrates the scenario within this architecture with two simultaneous frames where the frame from the sensor 1 is completed using a complete frame from the sensor 2.

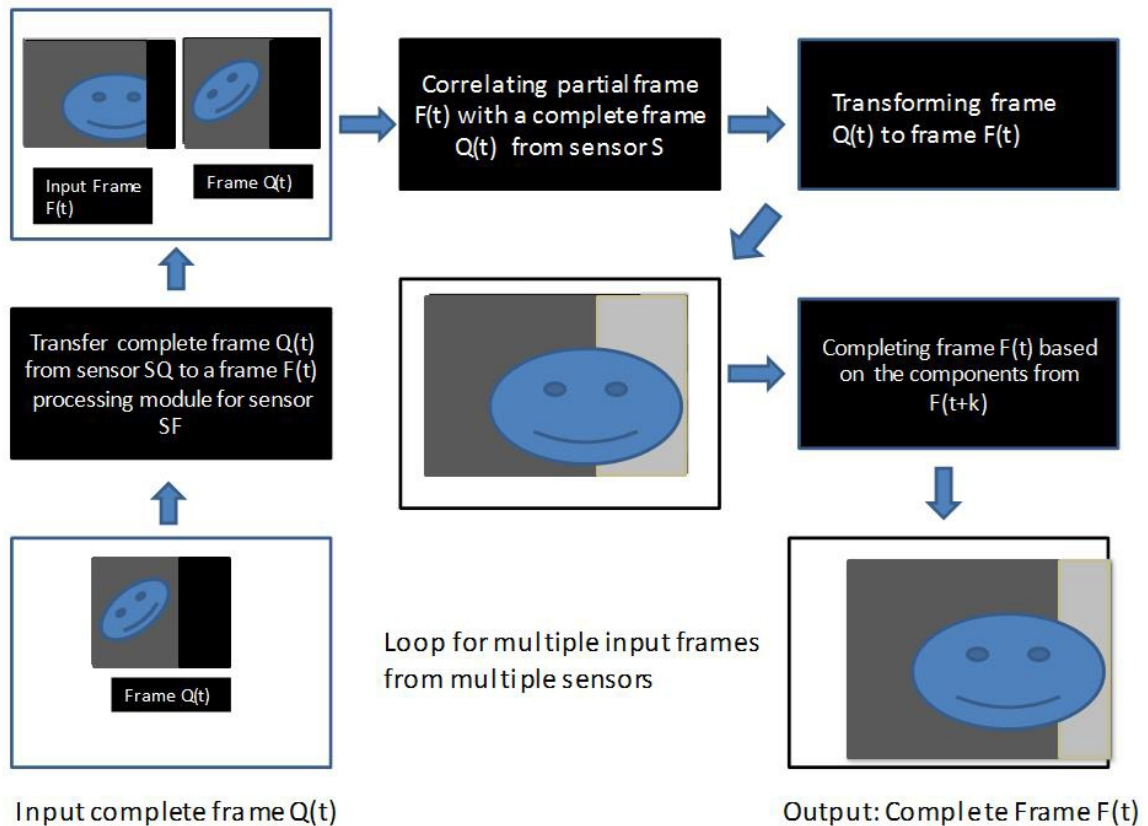


Figure 6. Two-sensor architecture for partial frame correlation and completion.

Architecture for two sensors at multiple times. The architecture for correlating partial frames from multiple sensors at multiple times is a combination of the architectures presented above. It is supporting the exchanging of frames between separate modules and then correlating frames from each sensor individually with possible multiple further exchanges of frames completed fully or partially by each computational module.

4. COMPUTATIONAL EXPERIMENTS

Experiment design. To evaluate the algorithm a testbed was developed that randomly generated frame images with small targets then steps 1-17 have been conducted to evaluate the algorithm on test data.

Step 1. Load frame image I, e.g., a 320x240 image with n small targets ($n \geq 10$) with 1 pixel size of each target, black background (B) and white small objects of pixels size (W), $I(320,240, n, B,W)$.

Step 2. Extract all n objects (points/pixels $\{P\}$) from image $I(320,240, n, B,W)$.

Step 3. Construct all triangles with points $\{P\}$ as vertices (a,b,c) and compute their centroids, $C = (a+b+c)/3 = ((x_a + x_b + x_c)/3, (y_a + y_b + y_c)/3)$, where $a = (x_a, y_a)$, $b = (x_b, y_b)$, $c = (x_c, y_c)$. Construct secondary triangles (triangles based on centroids) and compute centroids of secondary triangles that can be more robust.

Step 4. Compute areas of each triangle T ,

$$T = 0.5 \left| \det \begin{pmatrix} x_a & x_b & x_c \\ y_a & y_b & y_c \\ 1 & 1 & 1 \end{pmatrix} \right| = 0.5 |x_a y_b - x_a y_c + x_b y_c - x_b y_a + x_c y_a - x_c y_b|$$

Step 5. Compute a matrix M of all ratios of areas of triangles, T_i/T_j .

- Step 6. Find all triples $\{Q\}$ of elements outside of the diagonals that are similar in the two matrixes M_1 and M_2 for the two images from the two sensors. Most similar triples are identified by a threshold.
- Step 7. Find the closest (“best”) triple Q in $\{Q\}$.
- Step 8. Find an affine transform f based on Q .
- Step 9. Transform all extracted points by using f in the first image.
- Step 10. Evaluate the quality of match with the transform f .
- Step 11. Loop tasks 7-9 for other “best” triples Q to find the f that provides high accuracy.
- Step 12. Produce the raster and vector graphical outputs.

Experiments with partial frames with and without lost objects. Figure 7 shows the COPAF capabilities to correlate shifted frames that have no lost objects. All experiments with partial frames have been successful in finding the correct matches between objects in spite of rotation and rescaling. The correlated frames had a different number of objects (the second frame lost 20% of data at random). The automatic color-coding of triangles is used in Figures 8 and 9 to show the matches of triangles and objects. The matched triangles get the same color. This allows a quick visual verification of the correctness of the match. Figures 8 and 9 show the COPAF capabilities to correlate complete and a corrupted frame with 2 missing and 2 additional targets relative to other frames. Other conducted experiments had shown the abilities to correlate the corrupted frames with the 4 missing or the 4 additional targets relative to other frames.

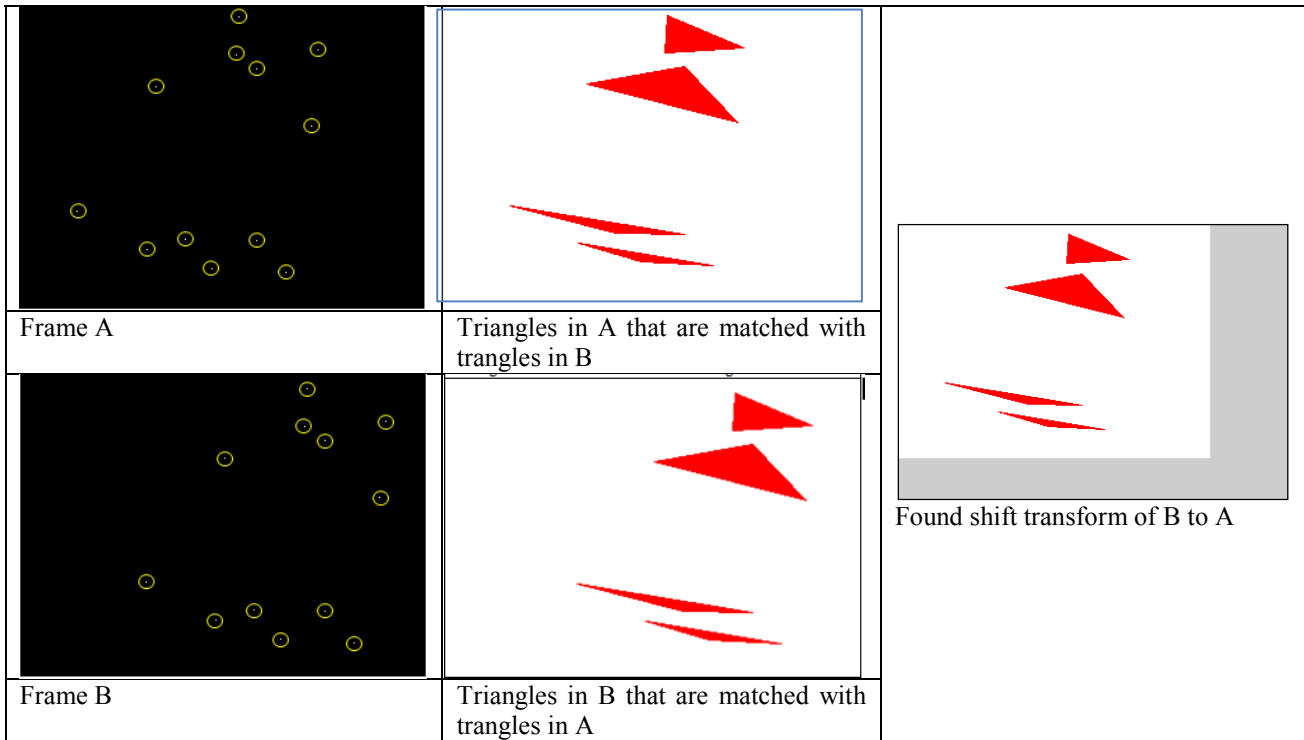


Figure 7. Finding transform with area ratio and triangulation that turned to be a shift of I2 to I1.

5. COMPUTATIONAL EFFICIENCY

Converting *raster data to the vector domain* permits fundamentally to speed up the computations with partial frames. One of the major time-consuming operations of the COPAF algorithm is generating multiple triangles and computing their areas.

Example. A right triangle with the sides of 100×100 pixels contains $100^2/2=5000$ pixels. Computing the centroid of this triangle in the raster domain requires identifying all these 5000 pixels and running a loop 5000 times to sum them up. In the vector domain we need just the three vertices a , b , and c and then use the simple formula shown above with two summations and one division without any looping.

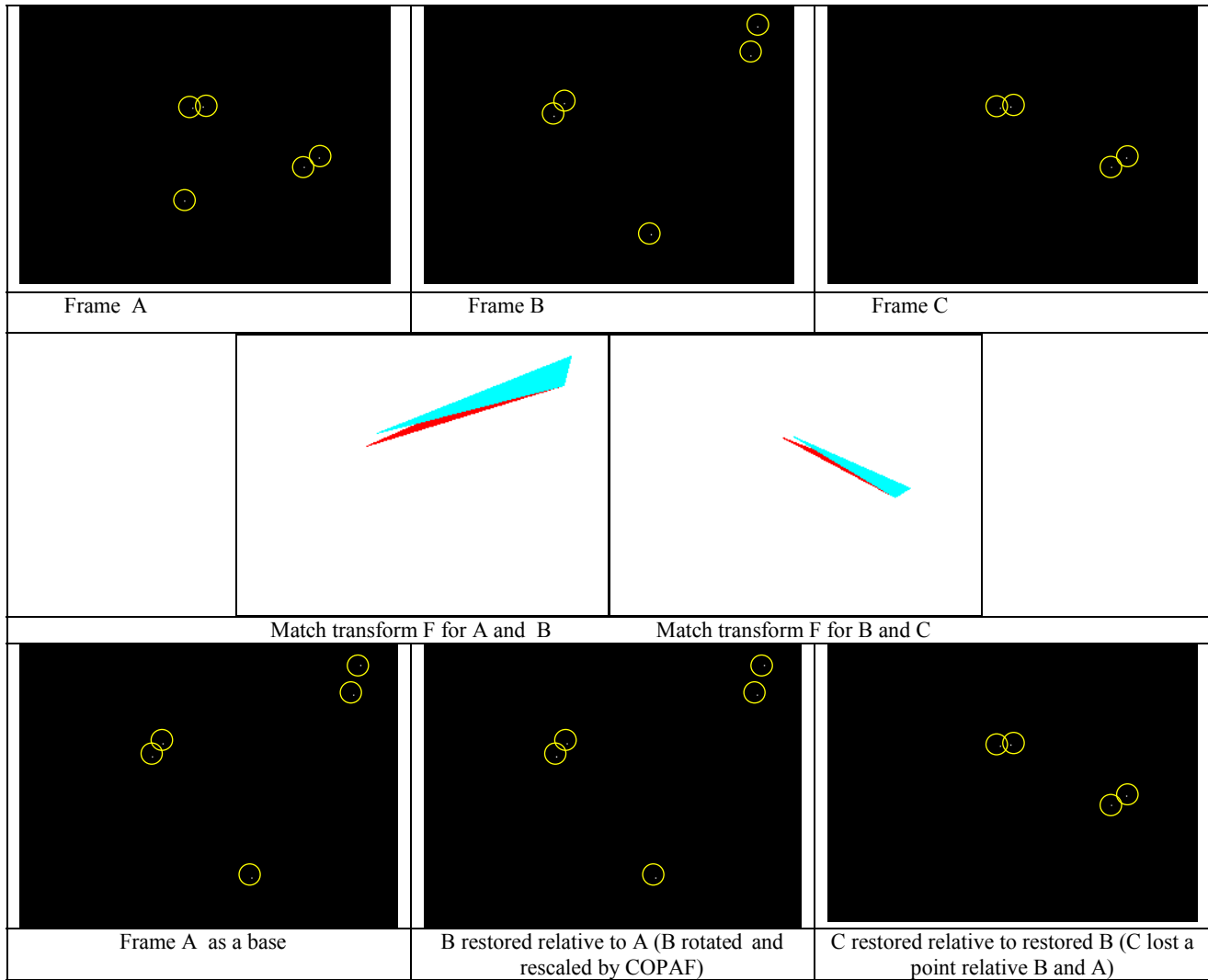


Figure 8. Automatic correlation of partial frames with automatic detection of matching elements, color-coding of matched triangles and constructing affine transforms.

In general for shapes that are more complex than a single triangle, triangulation is required. However, for the small (pixel-size) objects, the triangles are sufficient. To be able to use this advantage the extraction of objects from the image is required. For the non-uniform background, the problem is setting up an intensity threshold. For a “too low” threshold, we may get multiple false positive object’ pixels. For a “too high” threshold we may get multiple false negatives (target pixels that are missed) in the extraction process.

The major challenge of the COPAF task is to make its algorithm less computationally demanding. It has $O(n^4)$ computational complexity (n is the number of *possible triangles* in each frame) if search is conducted on all pairs of pairs of triangles from both frames. If the number of objects and respectively triangles is different in the two frames (n and m) then the complexity is $O(n^2m^2)$, which is no greater than $O(\max(n,m)^4)$. We have developed a new algorithm with the complexity of $O(r \cdot \max(n,m)^2)$. Here r is the number of pairs of triangles in the particular cover by triangles of all objects in the frame. This particular cover can also be minimized by taking triangles that have no common vertices in the frame with the lower number of targets.

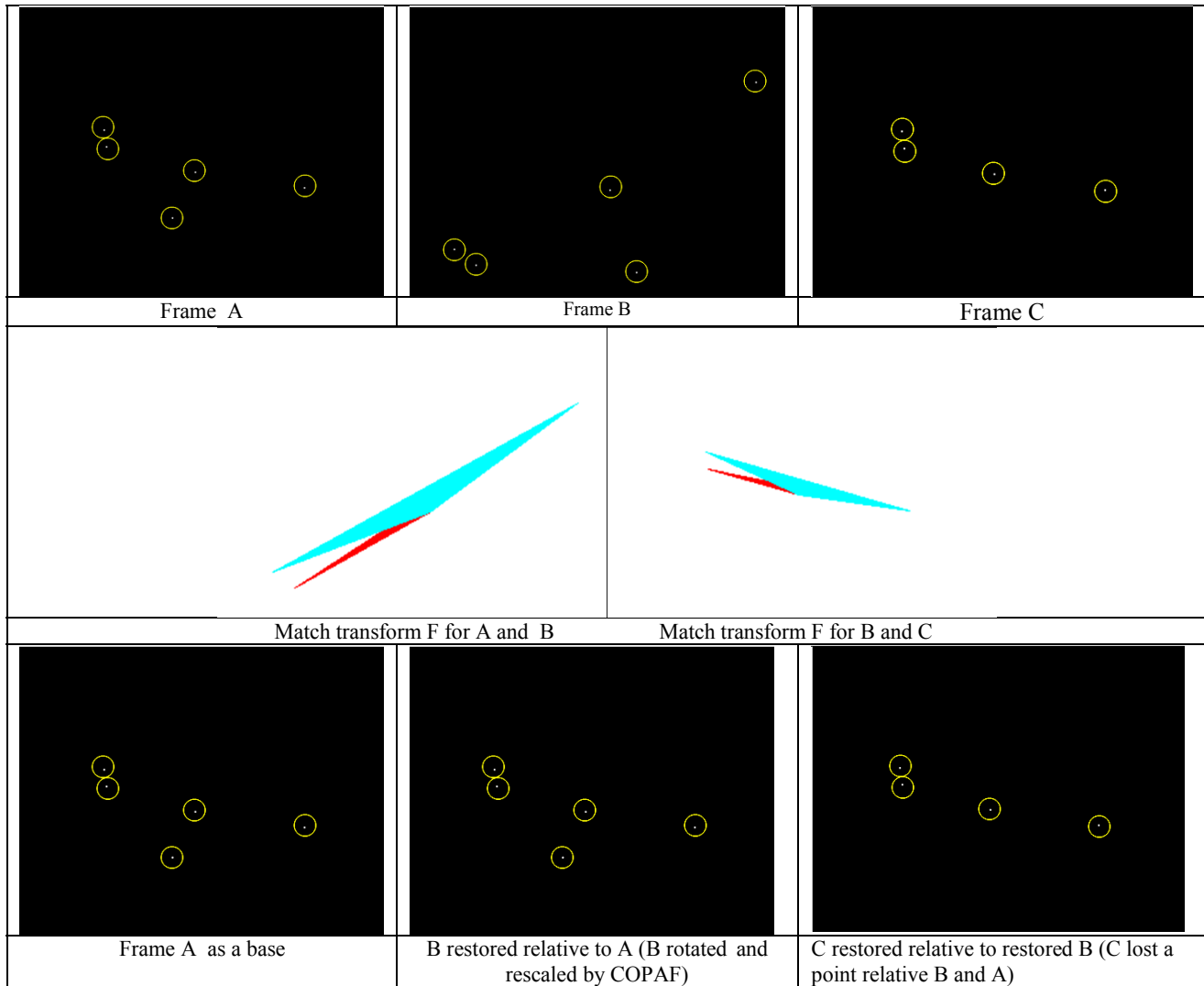


Figure 9. Automatic correlation of partial frames with automatic detection of matching elements, color-coding of matched triangles and constructing affine transforms.

6. CONCLUSION AND FUTURE WORK

This paper presented the architecture of the COPAF system and the results of the correlational computational experiments. These experiments had shown the feasibility of partial frame correlation with the proposed approach, the fully affine invariant COPAF algorithm. The COPAF is applicable for both objects of significant size as well as to objects of one-pixel size. Further computational experiments will help to clarify the COPAF algorithm's capabilities for frames with a high level of noise and low signal-to-noise ratio. The enhancing of the COPAF algorithm for the case when the blob extraction (segmentation) algorithms produce low quality inputs for the COPAF is another direction of the future work, along with raster vectorization and conducting COPAF completely in the vector domain, dramatically compressing data and decreasing execution time. These developments will create a base for the development of a full-scale system and the integration of it with the MapSnap System [Kovalerchuk et al, 2011] that fuses raster images and vector data. Approved for Public Release 13-MDA-7349 (14 June 13)

REFERENCES

- [1] Bar-Shalom Y., Blair W., (Eds.) *Multi-Target/Multi-Sensor Tracking: Applications and Advances*, Vol. III, Artech House, Norwood, MA, 2000.
- [2] Hall D., Llinas J. - An introduction to multisensor data fusion, *Proceedings of the IEEE*, vol. 85, No. 1, 1997
- [3] Kovalerchuk, B., Modeling ATR processes prior to experimentation to predict ATR system performance, *The 12th International Conference on Information Fusion*, Seattle, USA, 6-9 July 2009
- [4] Kovalerchuk, B., Structural and metric correlation of electro-optical and radar generated tracks, *Defense and Security Symposium, SPIE Proc. Vol. 6567-05*, 2007.
- [5] Kovalerchuk, B., Affine invariant and robust image registration/conflation algorithm, *SPIE Defense and Security Symposium, Orlando, FL, SPIE Proc, Vol., 6568, 6568-08*, 2007.
- [6] Kovalerchuk, B., Schwing J., (Eds). *Visual and Spatial Analysis: Advances in Visual Data Mining, Reasoning and Problem Solving*, Springer, 2005. 600 p.
- [7] Kovalerchuk, B., Doucette P., , Seedahmed G., Tagestad J., Kovalerchuk S., Graff B., *MapSnap System to Perform Vector-to-Raster Fusion*, *SPIE Proc*, 8053 – 5, 2011
- [8] Liggins M., Hall D., and Llinas J. (eds), *Handbook of Multisensor Data Fusion*, CRC, 2008
- [9] Zitova, B., Flusser, J. *Image registration methods: a survey*. *Image and vision computing*, 2003, 21(11), pp. 977-1000.
- [10] Morel J., Yu G., ASIFT, A new framework for fully affine invariant image comparison. *SIAM Journal on Imaging Sciences*, 2(2):438-469 (2009). DOI:10.1137/080732730
- [11] Yu, G., Morel J., "ASIFT: An Algorithm for Fully Affine Invariant Comparison." *Image Processing On Line* 2011 (2011). <http://dx.doi.org/10.5201/ipol.2011.my-asift>
- [12] K. Mikolajczyk, K. Tuytelaars T., Schmid C., Zisserman A., Matas J., Schaffalitzky F., Kadir T. and Van Gool, L, A comparison of affine region detectors, *International Journal of Computer Vision*, 65, 1-2 (2005). DOI:10.1007/s11263-005-3848-x
- [13] Paradowski, M., Śluzek A., Local keypoints and global affine geometry: Triangles and ellipses for image fragment matching, In: *Innovations in Intelligent Image Analysis*, Springer, 2011.
- [14] Paradowski M., Śluzek A., Automatic Visual Object Formation using Image Fragment Matching, *Proceedings of the International Multiconference on Computer Science and Information Technology* pp. 97–104, <http://proceedings2010.imcsit.org/pliks/52.pdf>
- [15] Pang Y., Lia W., Yuan Y., Pan J., Fully affine invariant SURF for image matching, *Neurocomputing*, Vol.85, 15 2012, pp. 6–10
- [16] Liu Z., Wang Y., Yu J., Zhao J., and Wang J., "An illumination and affine invariant descriptor for aerial image registration", *Proc. SPIE 8558, Optoelectronic Imaging and Multimedia Technology II*, 85580Z, 2012; doi:10.1117/12.999387; <http://dx.doi.org/10.1117/12.999387>
- [17] Kang, X., Han, C., & Yang, Y., Automatic SAR image registration by using element triangle invariants. In *Information Fusion, 2006 9th International Conference on* (pp. 1-7), IEEE, 2006
- [18] Hill F., Kelley S., *Computer Graphics Using OpenGL*, 3/E, Pearson, 2007
- [19] Lee Y-Y ; Kim C-S; Sang-Uk Lee, Video frame-matching algorithm using dynamic programming, *J. Electron. Imaging*. 18(1), 010504, 2009, doi:10.1117/1.3092367
- [20] Flusser J., T. Suk, A moment based approach to registration of images with geometric distortion, *IEEE Trans. Geosci. Remote Sensing*, 32(2), 1994, pp. 382-387.
- [21] Li Z., , Gao L., and Katsaggelos A. K., "Locally embedded linear subspaces for efficient video indexing and retrieval," *Proc. IEEE Intl. Conf. Multimedia Expo.* , pp. 1765–1768, 2006.
- [22] Yuan J., Wang W., , Meng J., Wu Y., , and Li D., "Mining repetitive clips through finding continuous paths.," *Proc. ACM Intl. Conf. Multimedia.* , pp. 289–292, 2007.
- [23] Delannay D., de Roover C., and Macq B., "Temporal alignment of video sequences for water-marking systems.," *Proc. SPIE. 0277-786X 5020*, 481–492, 2003.