VIDEO SUPER-RESOLUTION BASED ON LOCAL INVARIANT FEATURES MATCHING

Renan U. Ferreira[1], *Edson M. Hung*[2], *Ricardo L. de Queiroz*[3]

Universidade de Brasilia, Brasilia, Brazil
[1]Departamento de Engenharia Elétrica
[2]Faculdade do Gama - Engenharia Eletrônica
[3]Departamento de Ciência da Computação
E-mails: (renan,mintsu,queiroz)@image.unb.br

ABSTRACT

This paper presents an algorithm for video super-resolution based on scale-invariant feature transform (SIFT) matching. SIFT features are known to be a robust method for locating keypoints. The matching of these keypoints from different frames in a video allows us to infer high-frequency information in order to perform example-based super-resolution. We first apply a block constrained keypoint detection for a more precise superposition of features. Later, we extract high-frequency information with a gradient-based matching scheme. Our results indicate gains over interpolation and previous example-based super-resolution approaches.

Index Terms— Example-based super-resolution, Mixed-resolution video, Local invariant features, SIFT.

1. INTRODUCTION

Image super-resolution (SR) [1, 2] is the process in which an image has its resolution improved from inferred higher frequency information extracted from other images. In this process, the images should agree in some sense, either by belonging to a group of pictures of the same object (or similar content) or if their information match even if they are from completely different objects or scenes.

1.1. Example based video super-resolution

Example-based SR [3] is a process in which a group of reference images and their low-resolution versions are used to compose a database. For a given low-resolution image to be super-resolved, we search over the low-resolution database for a match (or region matches). Each match has an associated high-resolution image (or regions). From the low and high-resolution associated pair, we acquire the high frequency information to be applied to the original low-resolution image.

In example-based SR for mixed-resolution video, i.e. video made of frames in different resolutions (see illustration in Fig.1[1]), the database is composed by frames at high-resolution while the low-resolution ones are to be super-resolved after being upsampled back to the high (full) resolution. Mixed-resolution video is an approach to reduce complexity in video coding [4].

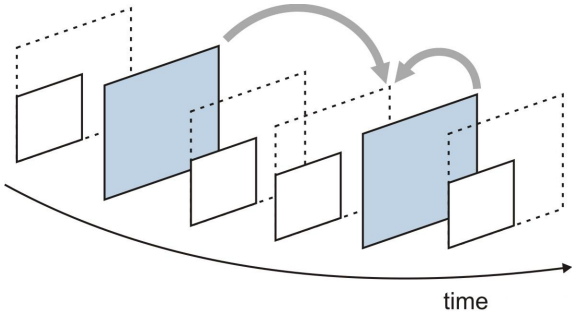


Fig. 1: Mixed-resolution video.

1.2. Local invariant feature

Local invariant features from images have been widely studied in the field of computer vision and used in various applications, such as object detection. In an image, a local feature is a pattern, such as points, edges or patches which differ from their immediate neighborhood. The feature is said to be invariant if it does not change over a certain transformation [5].

One of the most effective methods, since it has been presented by Lowe [6], is the Scale Invariant Feature Transform (SIFT). This method extracts features invariant to scale and rotation (fully scale invariant [7]). Besides, they are also robust to some changes in affine distortion, addition of noise and changes in 3D viewpoint and in illumination [8]. There is also a fully affine-invariant extension to SIFT, called ASIFT [9, 10]. In parallel, a brightness and contrast invariant and rotation-discriminating method robust to scaling and partial occlusion, called Fourier coefficients of radial projections (FORAPRO), has been presented [11], along with its affine extension, AFORAPRO, with significant improvements over ASIFT [12].

[1]The authors would like to thank Camilo C. Dorea for this figure.

2. SUPER-RESOLUTION THROUGH FEATURE MATCHING

The general idea of our work is to super-resolve low-resolution (LR) video frames through local invariant features matching in mixed-resolution videos. Let it be a mixed-resolution video composed by LR and high-resolution (HR) frames, such as in Figure 1. Let us now consider a single LR frame L and a HR frame H.

For our example-based SR, we will need two resampling functions: a downsampling function $d\{.\}$ which brings a frame from HR to LR; and an upsampling function $u\{.\}$ which brings a frame from LR to HR. It is of the most importance to remember that both functions require filtering in order to avoid aliasing effect.

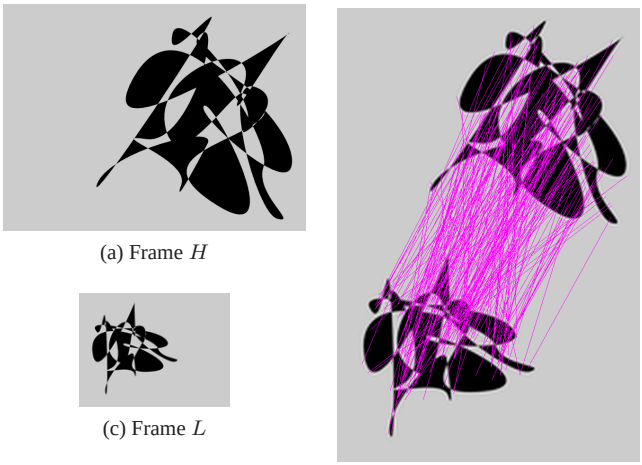
Since we want to super-resolve a video frame, it would be interesting put it in the same resolution of H frame. Being so, we obtain frame $L_u = u\{L\}$. For a comparison of information in the same frequency band, we derive frame $H_{ud} = u\{d\{H\}\}$. The idea now is to match the L_u and H_{ud} frames in order to infer the high-frequency information from the H frame. Through local invariant features, it is possible to match features from both upsampled frames and find a transformation $w\{.\}$ that warps one of them such that most matched points will be closely collocated [13].

Basically, our algorithm can be explained in a few steps. For two frames L and H, the first step is the calculation of frames L_u and H_{ud}. In step two, we apply local invariant features detection in both L_u and H_{ud}, match the keypoints detected and acquire the $w\{.\}$ transformation function. The third step is the calculation of frames $H_{wud} = w\{u\{d\{H\}\}\}$ and $H_w = w\{H\}$. A very common way, long used in image processing [14], to gather some high-frequency information from an image is to calculate the difference between itself and a blurred version of itself. This is has been done in SR [15,16] before and in our case, we estimate this high-frequency as $hf = H_w - H_{wud}$, which is our forth step. The fifth and last step is the calculation of L_{SR}, the super-resolved version of frame L, by a simple summation, i.e. $L_{SR} = L_u + hf$. We have noticed, however, that a little change in the algorithm can bring a great improvement to the results. In the fourth step, instead of using the blurred version of H_w as H_{wud}, we can use the frame $H_{udw} = u\{d\{w\{H\}\}\}$, i.e. we first warp the HR frame and only then we apply the resampling functions.

In Figure 2, we show an example of frames H, L (on the left side) and the matching between features of frames H_{ud} and L_u, on the right side. Next, in Figure 3, we see a zoom of the object from the same previous example in frames L_u, L_{SR} obtained with H_{wud} and L_{SR} obtained with H_{udw}.

3. BLOCK CONSTRAINED FEATURE MATCHING

Section 2 described the general idea of our algorithm. Unfortunately, video sequences are rarely composed of a single moving object. In this case, the simple matching of features acquired from the entire frames is not very effective. Figure 4 shows, as an example of this fact, in which frame 0 from sequence "mobile" is warped after being matched to frame 15. Note that in (b), the right end of the calendar spiral is now higher than the left one, compared to that in (a). This happens because, in this scene, while the camera moves left, the calendar moves upwards.



(a) Frame H

(c) Frame L

(b) Feature matching

Fig. 2: Frames H and L and matching between H_{ud} and L_u.

(a) L_u (b) L_{SR} with H_{wud} (c) L_{SR} with H_{udw}

Fig. 3: Zoom in L_u and different results of L_{SR}.

Aiming at the solution of this problem, we propose that the feature matching should be done in a block-wise fashion. First, we create a grid of blocks of an arbitrary fixed size. Then, we apply the grid to the L_u frame, so that the features of each block in the grid of this frame are matched to those of the entire H_{ud} frame separately. From this, there is a warping function generated for each block. We can then warp the H frame according to each function and operate the resampling functions. The warped versions of H will necessarily be the same size of the blocks. Because of that we can compose a new frame H_b with each H_w (one for each block) placed in its respective position in the grid. The same is done to the H_{udw} blocks, creating a new resampled frame H_r, i.e. a frame composed by the warped and resampled versions of H, correctly positioned over the grid. The resampling of each block in H_b prior to the composition of frame H_r (instead of resampling frame H_b) is important to avoid false blurred artifacts in the border of the blocks. Figure 5 (a) shows an example of a 128x128 pixels block from frame 15 of sequence

(a) Frame 15 (b) Frame 0 warped

Fig. 4: Bad frame warping.

(a) Block matching (b) H_b frame

Fig. 5: Good block warping and H_b frame composition.

"mobile" matched to frame 0. Figure 5 (b) shows the new resulting H_b frame for the specific grid to which this block belongs.

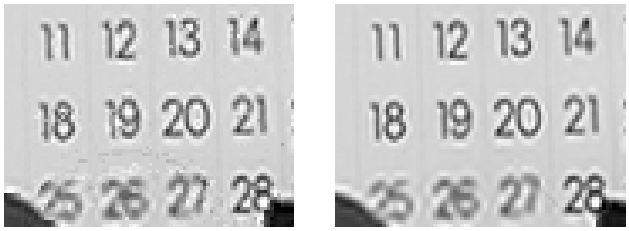
A single grid will not contemplate the correct matching of all objects and features. It is necessary then to vary both the position of the grid and the size of the blocks. Following the process explained above for each grid g, will give us two sets $\{H_b^g\}$ and $\{H_r^g\}$.

4. GRADIENT-BASED MATCHING

The last part of our algorithm is to acquire the best high-frequency information possible using the frames in the sets $\{H_b^g\}$ and $\{H_r^g\}$. Since our greatest interest is in the correct sharpening of contours, it is only natural to consider the use of gradients. Let us consider any given pixel $p_{i,j}$, located in position (i, j) of frame L_u and its gradient $\nabla p_{i,j}$, calculated by using Sobel gradient operators. Next, for of all the pixels $p_{i,j}^g$ in the same position (i, j) in all the frames in set $\{H_r^g\}$, we calculate the gradients $\nabla p_{i,j}^g$. For each position (i, j), the best match $\hat{g}_{i,j}$ will be the one that satisfies

$$\hat{g}_{i,j} = \underset{g}{\operatorname{argmin}} \|\nabla p_{i,j} - \nabla p_{i,j}^g\|. \quad (1)$$

Now that we have the best match in the L-2 norm sense, we compose a new frame X_l with the pixels $p_{i,j}^{\hat{g}}$. Since the two sets $\{H_b^g\}$ and $\{H_r^g\}$ are related, if $q_{i,j}^g$ is a pixel in frame H_b^g, we can also compose a new frame X_h with the pixels $q_{i,j}^{\hat{g}}$. Finally, our high-frequency information can now be calculated as $hf = X_h - X_l$.



(a) Single gradient (b) SAD of gradients

Fig. 6: Comparison of gradient.

Unfortunately, this matching of gradients is subject to many sorts of problems ranging from a bad warping function, generating a wrongly warped frame, to block borders discontinuities. In order to minimize this problem, we use the matching of the gradients of pixels $p_{i,j}$ and others in a neighborhood around it in a SAD (sum of absolute differences) fashioned way. For a square window of size $2m + 1$ around pixel $p_{i,j}$, our new best match $\hat{g}_{i,j}$ must now satisfy

$$\hat{g}_{i,j} = \underset{g}{\operatorname{argmin}} \sum_{k=-m}^{m} \sum_{l=-m}^{m} \|\nabla p_{i+k,j+l} - \nabla p_{i+k,j+l}^g\|. \quad (2)$$

Clearly, the previous case of equation (1) is a restriction of this one, when $m = 0$. This SAD of gradients procedure showed some robustness in the final super-resolution result, specially in eliminating granularity noises over occlusion regions. We show an example of the different results in the final SR process in the two cases, with $m = 0$ and $m = 7$, in Figure 6 (a) and (b), respectively.

5. EXPERIMENTAL RESULTS

In this section we describe details of our implementation, experiments and results. First of all, Lanczos-3 was the linear filter used in the resampling function for both down and upsampling. The local invariant feature method chosen was SIFT. Despite the fact that other methods report better results, as mentioned in Subsection 1.2, we opted for using an open-source SIFT library [2]. This library brings some very useful implementations, such as the detection and matching of features, the derivation of the warping function from the matched features and the application of the warping itself. We only needed to apply some minor changes for our specific tests.

In the block matching step, we used square blocks of sizes 64, 128 and 256 pixels. In the case of the 64x64 blocks, the grid displacements were in steps of 32 pixels, both in the vertical and horizontal directions. For the other sizes, the displacements were in a value equal to the block size divided by 16, i.e. steps of 8 and 16 for block sizes 128 and

[2]Downloaded from http://blogs.oregonstate.edu/hess/code/sift/, on July 28th, 2011. Details in [13]

256, respectively. In order to test the impact of the block sizes, we composed independent sets for each size. As for the gradient-based matching step, we tested window sizes with $1 \leq m \leq 15$.

The validation of our algorithm is tested with four CIF and two 720p size sequences. For each sequence, we super-resolve the LR version of the 16th frame (acquired from the downsampling of the original frame by a factor of two) using the 1st and the 31st ones as reference HR frames. For the 720p sequences, this counting starts from the first non-grey frame.

We compare our results to the works described in [17] and [18]. We directly compare ours to those reported for the two different methods in [17]: an example-based motion estimation SR (MSR) and a hybrid (HSR) that combines MSR with an on-the-fly training dictionary. As for the comparison to [18], we had to run new tests. We used the reference frames as training sets and the dictionary used to super-resolve the frame was composed by 1000 patch-pairs. It is important to mention that the approach in [18] is not an example-based SR method, but it assumes that the super-resolved image is a sparse representation of raw patches. Since their implementation was made available, we considered it would be important to compare our results.

Table 1 shows all the comparisons of results. The column named "Lanczos" shows the distortion caused by the downsample/upsample process. Our results shown here were obtained with a configuration that, on average, lead to the best distortion values. This configuration was the use of 128x128 blocks in the block-constrained feature matching step and $m = 6$ in the gradient-based matching step. We achieved an average improvement of 2.8 dB over the other methods.

Table 1: PSNR [dB] comparison among SR and interpolation methods.

Sequence	Bicubic [17]	Lanczos	SR in [18]	MSR [17]	HSR [17]	Our SR
Container	27.9	27.4	30.7	31.9	33.2	**35.0**
Hall	29.1	28.2	32.6	37.4	38.0	**40.3**
Mobile	22.9	22.8	25.5	24.5	25.5	**28.6**
News	29.4	30.1	34.1	31.9	36.1	**39.1**
Mobcal	27.7	27.8	29.8	30.9	31.0	**36.1**
Shields	31.1	33.1	34.9	31.4	32.7	**36.4**

Some other configurations showed slightly better results, depending on the sequence. The largest variation from the result shown in Table 1 was observed for sequence "hall". Using the same value of $m = 6$, but blocks of size 256, we reach a distortion of 40.6 dB (instead of 40.3 dB).

6. CONCLUSIONS

We have presented a new example-based video super-resolution method through invariant local feature matching (using the SIFT method). Block-constrained SIFT features matching was shown to be a good solution for independent contour overlap. A SAD-fashioned gradient-based matching was used and applied to super-resolving frames in mixed-resolution sequences. Results show that the method can outperform other existing methods of example-based super-resolution on an average of 2.8 dB.

7. REFERENCES

[1] S. Chaudhuri, *Super-Resolution Imaging*, Kluwer, 2001.

[2] A. K. Katsaggelos, R. Molina, and J. Mateos, *Super Resolution of Images and Video*, Synthesis Lectures on Image, Video, and Multimedia Processing. Morgan and Claypool Publishers, 2007.

[3] W. T. Freeman, T. R. Jones, and E. C. Pasztor, "Example-based super-resolution," *IEEE CGA*, vol. 22, pp. 56–65, March 2002.

[4] D. Mukherjee, B. Macchiavello, and R.L. de Queiroz, "A simple reversed complexity wyner-ziv video coding mode based on a spatial reduction framework," in *Proc. SPIE Visual Commun. and Image Process.*, January 2007.

[5] T. Tuytelaars and K. Mikolajczyk, "Local invariant feature detectors: A survey," *Found. and Trends in Comput. Graph. and Vision*, vol. 3, no. 3, pp. 177–280, 2008.

[6] D.G. Lowe, "Object recognition from local scale-invariant features," in *Proc. ICCV'99*, Corfu, Greece, September 1999, pp. 1150–1157.

[7] J.M. Morel and G.Yu, "Is sift scale invariant?," *Inverse Problems and Imaging*, vol. 5, no. 1, February 2011.

[8] D.G. Lowe, "Distinctive image features from scale-invariant keypoints," *Intl. Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, January 2004.

[9] G. Yu and J.M. Morel, "A fully affine invariant image comparison method," in *Proc. ICASSP'09*, Taipei, Taiwan, April 2009.

[10] J.M. Morel and G.Yu, "ASIFT: A new framework for fully affine invariant image comparison," *SIAM Journal on Imaging Sciences*, vol. 2, no. 2, pp. 438–469, 2009.

[11] H.Y. Kim, "Rotation-discriminating template matching based on fourier coefficients of radial projections with robustness to scaling and partial occlusion," *Pattern Recognition*, vol. 43, pp. 859–872, 2010.

[12] G.A.P. López and H.Y. Kim, "Novo algoritmo para reconhecimento de objetos invariante afim," in *Proc. SBrT'11*, Curitiba, Brazil, October 2011, (in Portuguese).

[13] R. Hess, "An open source SIFT library," in *Proc. ACMMM'10*, Firenze, Italy, October 2010.

[14] Rafael C. Gonzalez and Richard E. Woods, *Digital Image Processing (3rd Edition)*, Prentice-Hall, Inc., NJ, USA, 2006.

[15] F. Brandi, R. de Queiroz, and D. Mukherjee, "Super resolution of video using key frames," in *Proc. ISCAS'08*, Seattle, USA, May 2008.

[16] K.F. Oliveira, F. Brandi, E.M. Hung, R.L. de Queiroz, and D. Mukherjee, "Bipredictive video super–resolution using key–frames," in *Proc. IS&T/SPIE Symp. on Electron. Imaging, VIPC*, San Jose, USA, January 2010.

[17] B. C. Song, S.-C. Jeong, and Y. Choi, "Video super-resolution algorithm using bi-directional overlapped block motion compensation and on-the-fly dictionary training," *IEEE Trans. CSVT*, vol. 21, no. 3, pp. 274–285, 2011.

[18] J. Yang, J. Wright, T.S. Huang, and Y. Ma, "Image super-resolution as sparse representation of raw image patches," in *Proc. CVPR'08*, June 2008, pp. 1–8.