Hot or Not: Exploring Correlations Between Appearance and Temperature

Daniel Glasner
Harvard University

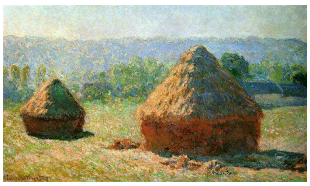
Pascal Fua
EPFL

Todd Zickler
Harvard University

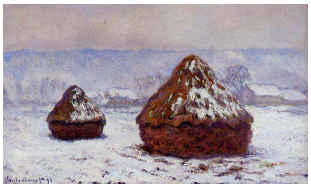
Lihi Zelnik-Manor
Jacobs-Technion-Cornell

Abstract

In this paper we explore interactions between the appearance of an outdoor scene and the ambient temperature. By studying statistical correlations between image sequences from outdoor cameras and temperature measurements we identify two interesting interactions. First, semantically meaningful regions such as foliage and reflective oriented surfaces are often highly indicative of the temperature. Second, small camera motions are correlated with the temperature in some scenes. We propose simple scene-specific temperature prediction algorithms which can be used to turn a camera into a crude temperature sensor. We find that for this task, simple features such as local pixel intensities outperform sophisticated, global features such as from a semantically-trained convolutional neural network.



(a) Claude Monet, *'Grainstacks at the End of Summer, Morning Effect'*



(b) Claude Monet, *'Grainstacks, Snow Effect'*

(c) hot...

(d) or not...

Figure 1: How do we perceive temperature differences?

1. Introduction

Great painters are able to convey properties of the environment through their artistic expression. Monet's painting[1] in Figure 1a conveys a sense of summer, whereas the one in Figure 1b captures the essence of winter. While most of us cannot produce such wonderful paintings, we do share the ability to perceive the temperature differences. We can also do this easily with natural images, such as those shown in Figures 1c, 1d. This paper explores which visual aspects of an outdoor scene correlate with the ambient temperature.

Changing the temperature in a controlled laboratory environment will usually not have a direct, easily measurable, effect on the appearance of an object. However, ambient temperature and the appearance of an outdoor scene are highly correlated through multiple complex interactions. For example, the presence or absence of clouds, the angle of the sun, precipitation, and the color of leaves are examples of visually measurable attributes which offer indications about which temperatures are more or less likely.

Our focus in this paper is on understanding interactions between visual elements and the temperature. We adopt a statistical approach and study correlations between image sequences captured by static outdoor cameras over periods of time and temperature measurements from their vicinity.

In Section 3 we identify aspects of the image sequences, which are more indicative of temperature. As a way to assess the utility of such correlations in Section 4, we develop and test a number of predictive models. These models are trained in a supervised fashion from a time-lapse sequence of a scene, and they are tested according to their ability to predict temperatures from images of the same scene that are captured months or years later. The algorithms we compare make assumptions which would prevent them from working "in the wild". They are not meant to be competitive, but rather to lend insight into the problem.

Image sequences from statically mounted outdoor cameras usually exhibit some motion. In order to study these sequences we had to compute transformations to align and stabilize them . Stabilizing the sequences led us to an unexpected observation, in some sequences there is significant correlation between the motion of the camera and the temperature. We conjecture that these correlations can be explained by contraction and expansion of the camera mount.

To summarize, our contributions are:

- An analysis of correlations between image sequences

[1] Public domain Monet images from www.wikiart.org.

and temperature for outdoor scenes.

- A dataset of time-lapse sequences and the corresponding temperature measurements.
- Evaluation of algorithms for temperature estimation for a specific scene and time of day.
- A robust method to align sequences with significant inter-frame variation (described in Appendix A).

1.1. Related Work

In a sequence of papers of impressive breadth and depth (e.g. [12]) Narasimhan, Nayar and coauthors study the weather in outdoor images. They adopt a physics-based approach and devise models for the effect of weather on images and utilize them to solve various computer vision problems. In [13] Narasimhan et al. introduce the WILD dataset consisting of registered and calibrated images of a fixed outdoor scene under a range of weather conditions.

In 2006 Jacobs, Pless [7] and collaborators started collecting images in the Archive of Many Outdoor Scenes (AMOS). By now, the dataset contains more than half a billion images. The dataset is a long-term archive of imagery from publicly accessible outdoor webcams around the world. Using imagery from this dataset Jacobs et al. [6, 5] learn to predict wind velocity and vapor pressure from webcam images in a supervised setting. Islam et al. [4] collect and align weather data to webcams in the AMOS dataset, and explore its utility in predicting scene appearance.

Laffont et al. [8] study the appearance changes of outdoor scenes. They identify and explore high level properties termed "Transient Attributes" which affect scene appearance, such as "snow", "autumn" and "fog". They train regressors to estimate these attributes and demonstrate an image editing application in which they synthesize the appearance of a scene under different weather conditions.

Finally, Lu et al. [10] introduce a labeled dataset and a method for binary classification of images as sunny or cloudy. They propose a 'weather feature' consisting of five components: sky, shadow, reflection, contrast and haze.

2. Data Collection and Processing

We collect time-lapse sequences of images from outdoor webcams. For each webcam we find a nearby weather station and retrieve the corresponding historic temperature measurements.

Images: We download images from the AMOS dataset [7]. This dataset contains archived photos from outdoor webcams found on the internet. We manually selected 10 relatively stable scenes using the interface's search engine with the tag 'possibly stable' from various locations in the U.S.A. From each webcam we extract one image every day at 11:00 am local time over a period of two consecutive years. We refer to the cameras as scenes (a)-(j)[2].

Temperature measurements: We collect historic temperature measurements from data provided by "NOAA Cooperative Network" and obtained from the "Midwestern Regional Climate Center, cli-MATE"[3]. To identify the closest weather station to a scene we use the coordinates associated with the AMOS geolocation tag.

Preprocessing: We automatically discard images for which no temperature measurements were found, corrupt image files and 'stuck' frames, ones in which there is no change from the previous frame. After this filtering, the average number of valid frames per year is just over 300 and the total number of frames is approximately 6000. We resize all images such that their larger dimension is 320 pixels. Despite being on static mounts, the image sequences exhibit small, and occasionally large motion. We align the image sequences to compensate for these motions. As noted in [6] aligning outdoor webcam data can be challenging. Since standard alignment techniques proved inadequate, we propose a new robust alignment method described in Appendix A.

We will make the dataset publicly available.

3. Image Sequences and Temperature

In this section we examine correlations between image sequences and temperature measurements. We observe two types of correlations, between pixel intensities and the temperature and between camera motion and the temperature.

3.1. Correlating Intensities and Temperature

We start our analysis by examining the interactions between single pixel measurements and the temperature. The benefit of this simple approach is that it allows us to identify and understand the behavior at specific regions of the scene.

Let $T(t)$ be the temperature measured at time $t \in \mathcal{T}$ and $x_{i,j,c}(t)$ the sequence of measurements at pixel i, j and color channel c at time $t \in \mathcal{T}$. $\mathcal{T}$ consists of one time point close to 11:00 am local time, for every day of the year (excluding days filtered in the preprocessing).

For each scene, we use a year's worth of data to compute an ensemble of pixel-wise regressions, with one regression for each pixel i, j and color channel c. Each regression of the ensemble yields estimated temperatures $\hat{T}_{i,j,c}(t)$ for that scene. We evaluate these estimates using the *coefficient of determination* $R^2 \in [0, 1]$

$$R^2(\hat{T}(t), T(t)) = 1 - \frac{\sum_{t\in\mathcal{T}}(\hat{T}(t) - T(t))^2}{\sum_{t\in\mathcal{T}}(\bar{T}(t) - T(t))^2} \quad (1)$$

[2]The corresponding AMOS camera indices are 17603, 156, 204, 4181, 484, 90, 338, 842, 4556, 15767.

[3]http://mrcc.isws.illinois.edu/CLIMATE/

where $\bar{T} = \frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}} T(t)$. The coefficient of determination can be understood as the fraction of the variance in T that is explained by the estimator $\hat{T}$ in a linear regression. For a linear regression the coefficient of determination coincides with the square of the *sample correlation coefficient*, thus it can be understood as a measure of correlation.

Using R^2 to evaluate the estimates lets us compare results for different scenes using the same units. Another measure of interest is the Root Mean Squared Error (RMSE)

$$\text{RMSE}(\hat{T}(t), T(t)) = \sqrt{\frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}}(\hat{T}(t) - T(t))^2}. \quad (2)$$

While RMSE isn't directly comparable across scenes it provides an intuitive interpretation of the estimate's quality.

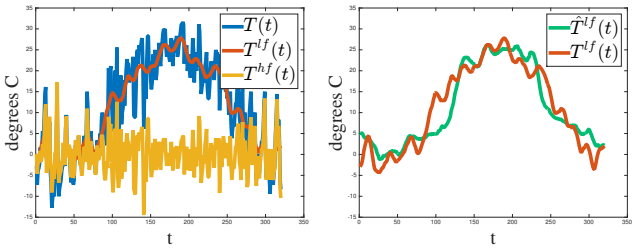
The first row of Table 1 summarizes the maximal R^2 values and the corresponding minimal RMSE values computed for one year from each of our scenes. The maximum is taken over all pixels and all color channels. We can see that for most scenes there are pixels whose intensities are fairly well correlated with temperature. Next, we show that by applying a temporal frequency decomposition we can achieve even stronger correlations.

3.1.1 Frequency Decomposition

Looking at a graph of the temperature over a year such as the one shown in Figure 2a we observe a distinctive low frequency component corresponding to the season and a high frequency component which corresponds to the day-to-day weather fluctuations. Motivated by this observation we decompose our time varying signals into a low-frequency and a high-frequency component in order to examine the correlations between these components separately. The decomposition is given by $T(t) = T^{lf}(t) + T^{hf}(t)$ and $x_{i,j,c}(t) = x^{lf}_{i,j,c}(t) + x^{hf}_{i,j,c}(t)$ where $s(t)^{lf} = s(t) * k$ is the *temporal* convolution of the signal $s(t)$ with a Gaussian kernel with $\sigma = 5$ days. An example decomposition for the temperature from the scene shown in column (e) of Figure 3 can be seen in Figure 2a.

We repeat the regression process as before, this time estimating the low-frequency temperature from the low-frequency intensities and the high-frequency temperature from the high-frequency intensities. The maximal (over i, j, c) R^2 values are summarized in the second and third rows of Table 1. We can see that in *all of the scenes* there are pixels whose intensities are *very good* estimators of the low-frequency temperature (the season). Estimating the high-frequency temperature is much harder. However, by summing the low- and high-frequency estimates we can produce a new improved estimator of the temperature $\hat{T}^{lf} + \hat{T}^{hf}$. The R^2 values for this estimator are in row 4 of Table 1.

To get a better sense of what these values mean we plot in Figure 2b an example of the low frequency temperature of



(a) Decomposition: low and high temporal frequency components.

(b) Low temporal frequency of the temperature and its estimate.

Figure 2: Frequency decomposition of temperature.

the scene shown in column (e) of Figure 3 and its estimate using a pixel from the area marked in red in column (e) of Figure 3. The average of the (signed) error is $0°$ and the standard deviation is $2.5°$C ($4.5°$F).

3.1.2 Understanding Correlations

Since we've computed fits from single pixels it is easy to visualize the results and identify highly correlated regions. The $R^2 \in [0, 1]$ values can be visualized as a color image. Figure 3 shows visualizations of $R^2(\hat{T}^{lf} + \hat{T}^{hf}, T)$ for the first year of each of the scenes along with an example image of each scene.

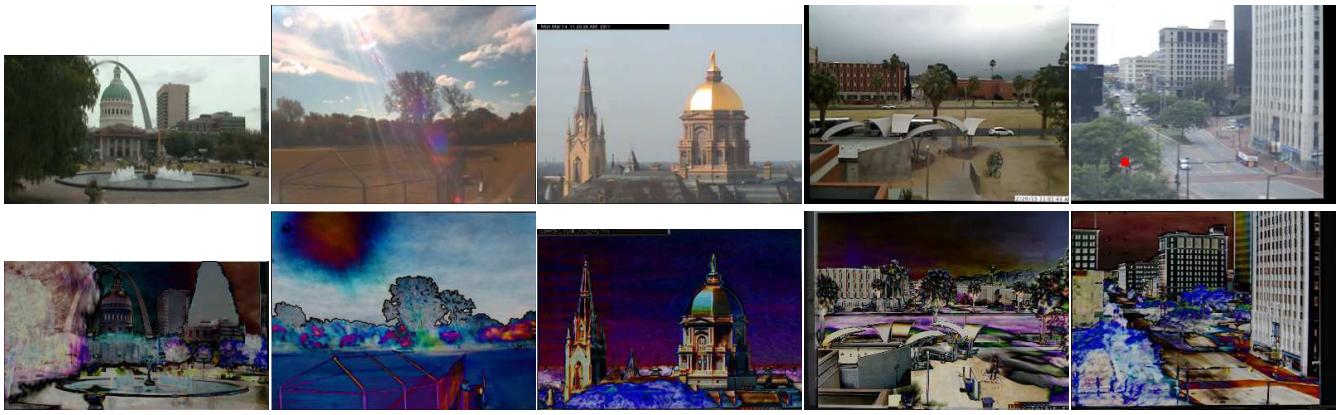
We interpret a dark area in the R^2 image as an area which does not correlate well with the temperature. A white or bright grey area is one in which the intensities in all color channels are highly correlated with the temperature and a color region suggests that the intensities in certain color channels correlate more with the temperature than others.

Looking at Figure 3 we can see that regions which are highly correlated with temperature are often semantically meaningful. We make the following observations

- Certain reflective oriented surfaces exhibit strong correlations. Surfaces such as roads, and building facades reflect light at an intensity that depends on the surface's material properties and on its angle with respect to the sun. Thus, the reflected intensity, as measured by the camera, is correlated with factors such as the sun's position in the sky (higher in summer and lower in winter) and on the presence or absence of clouds. Highly correlated surfaces can be seen in columns (a), (c)-(g), (i)-(j). Looking at column (e) we can note the effect of the combination of material and orientation on the correlation. The two walls of the building closest to the camera are made of the same material but one is parallel to the sun's path while the other is perpendicular. The parallel wall correlates with the temperature while the perpendicular one does not. A similar phenomenon can be seen on the different facets of the dome in (c).

	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)
$R^2(\hat{T}, T)$	0.69 / 5.45	0.54 / 8.02	0.53 / 7.70	0.69 / 5.01	0.67 / 6.37	0.67 / 2.60	0.73 / 5.32	0.65 / 6.29	0.37 / 3.17	0.61 / 6.89
$R^2(\hat{T}^{lf}, T^{lf})$	0.91 / 2.63	0.88 / 3.70	0.95 / 2.32	0.93 / 2.14	0.93 / 2.55	0.92 / 1.12	0.94 / 2.21	0.92 / 2.90	0.83 / 1.21	0.90 / 3.26
$R^2(\hat{T}^{hf}, T^{hf})$	0.31 / 3.55	0.11 / 4.41	0.16 / 3.59	0.23 / 2.90	0.19 / 4.05	0.22 / 1.94	0.25 / 3.38	0.11 / 3.10	0.28 / 2.14	0.14 / 3.54
$R^2(\hat{T}^{lf} + \hat{T}^{hf}, T)$	0.73 / 5.05	0.73 / 6.15	0.83 / 4.61	0.79 / 4.10	0.78 / 5.27	0.69 / 2.51	0.81 / 4.51	0.82 / 4.50	0.51 / 2.81	0.78 / 5.16

Table 1: **Correlating temperatures and intensities:** for each scene, we compute an ensemble of pixel-wise linear regressions between the time varying intensities (and their temporal frequency components) and the temperature. We report the best result achieved by regressing from a single pixel for each scene. Each cell contains two values, R^2 (coefficient of determination) and Celsius Root Mean Squared Error, (higher R^2 and lower RMSE is better).



(a) St. Louis, MO (b) Wheeling, IL (c) Notre Dame U, IN (d) Tucson, AZ (e) Akron, OH

(f) San Fransisco, CA (g) Newark, NJ (h) Muskegon, MI (i) Los Angeles, CA (j) Montrose, CO

Figure 3: **Interpreting correlations:** for each scene we show a representative frame and a 3-channel R^2 image which shows how well each pixel from each color channel correlates with the temperature (stronger intensities mean more correlation).

- There are strong correlations in foliage regions. This makes sense since leaves of deciduous trees are present in the summer when temperatures are warmer but absent in the winter when it gets colder. Such correlations can be observed in columns (a)-(b), (e), (g) and (j). It's interesting to note that palm trees which are present in columns (d) and (i) or evergreens more generally will not exhibit these correlations.

- Contrary to our expectation we found that sky regions are not consistently indicative of the temperature. An exception is seen in column (b) where the sun is directly visible in the frame.

- The scene shown in column (i) exhibits the lowest correlations. This scene shows a beach in Los Angeles where weather fluctuations are not as dramatic. The appearance of the scene does not change much throughout the year.

3.2. Correlating Motion and Temperature

Aligning the image sequences, as part of the data preprocessing, we discovered that for some of the scenes there is significant correlation between the computed camera mo-

tion and the temperature. We conjecture that these correlations can be explained by contraction and expansion of the camera mount due to temperature variation. Such contractions and expansions should be consistently aligned with the motion in the image sequence, as well as with the temperature. Another plausible contributing factor is the wind, however, wind measurements retrieved from some of the weather stations accounted only for a small part of the observed correlation.

We model our frame-to-frame alignments using homographies. Let $\{H_{t,k}\}_{t\in\mathcal{T}}$ be a sequence of transformations from frame t to some reference frame k. We characterize each transformation in the sequence by its action on a reference point $(1,1,1)^T$. Let

$$(m_x(t), m_y(t), 1)^T \propto H_{t,k}\,(1,1,1)^T \tag{3}$$

We report the values $R^2(m_x(t), T(t))$ and $R^2(m_y(t), T(t))$ in Table 2 along with the R^2 values of estimators computed from their low-frequency versions.

For many of the sequences, at least one of the directions of motion exhibits a non-negligible amount of correlation with the temperature. In sequences (b), (f), and (h) the low frequency motion accounts for *more than* 60% *of the variance* in the temperature signal. In such highly correlated instances the temperature might prove useful in applications, for example as a prior for sequence alignment.

4. Making Predictions

In this section we explore potential applications for the correlations we observed. First, we train algorithms to estimate the temperature from intensities. We then explore the flip-side of the correlation and synthesize an image with the appearance of a given temperature.

The algorithms we discuss in this section are scene and time of day specific, they assume a fixed viewpoint at approximately the same time of day. Some of the algorithms also assume that the images are well aligned. These assumptions clearly prevent the algorithms from being competitive or working "in the wild". Nevertheless, we chose to explore this setting in order to gain insight into the relationship between visual properties and the temperature with a minimum of distracting factors.

4.1. Predicting Temperature

We train algorithms to estimate the temperature for previously unseen frames from the same camera in a supervised learning setting. For each scene we use a set of images from one year as training data and images from the following year as testing data. All images are aligned.

We now describe 5 different estimation methods, the first three use simple pixel intensities as features while the last two use more sophisticated global image features. Based on our observations from Section 3.1.1 we apply the first three estimation methods independently to the low-frequency and to the high-frequency components of the pixel intensity sequences. At each pixel we sum the estimates of the two frequency components to produce an estimate of the temperature from that pixel. We denote the training (testing) intensities, temperature and time samples with x, T and $\mathcal{T}$ (x', T' and $\mathcal{T}'$) respectively.

Local Regression (LR) Here we try to estimate the temperature using single pixels as predictors. For each time series $x(t) = x_{i,j,c}(t)$ in our training data we compute regression coefficients $a^* = a^*_{i,j,c}, b^* = b^*_{i,j,c} \in \mathbb{R}$ by solving

$$a^*, b^* = \min_{a,b} \sum_{t\in\mathcal{T}} \left(ax(t) + b - T(t)\right)^2. \tag{4}$$

Thus, each pixel i, j, c gives us a prediction of the test temperatures $\hat{T}'_{i,j,c}(t),\ \forall t \in \mathcal{T}'$ by computing $\hat{T}'_{i,j,c}(t) = a^*_{i,j,c}x'_{i,j,c} + b^*_{i,j,c}$. To produce a single prediction we combine the predictions from all of the pixels using a weighted average based on the quality of the fit during training. Let $w_{i,j,c} = R^2(T, \hat{T}_{i,j,c})$ we set

$$\hat{T}'(t) = \sum_{i,j,c} \frac{w^p_{i,j,c}\hat{T}'_{i,j,c}(t)}{\sum_{i,j,c} w^p_{i,j,c}} \tag{5}$$

where $p \in \mathbb{R}$ is a power which characterizes the rate of decay of the coefficients. In all our experiments we use $p = 10$ (the algorithm is insensitive to this choice, other high powers produce similar results).

Local Regression with a Temporal Window (LRTW) Here again, we generate estimates using a single pixel but for each pixel we also use past measurements. We use a temporal window of size $W = 9$ and compute regression coefficients $a^*_{i,j,c} \in \mathbb{R}^W$ and $b^*_{i,j,c} \in \mathbb{R}$

$$a^*, b^* = \min_{a,b} \sum_{t\in\mathcal{T}} \left(\sum_{s=0}^{W-1} a(s)x(t+s) + b - T(t)\right)^2. \tag{6}$$

Per-pixel predictions are computed using the corresponding coefficients $a^*_{i,j,c}, b^*_{i,j,c}$ and combined to produce a single estimate $\hat{T}'(t)$ as in Equation (5).

Global Regularized Regression (GRR) Of course we don't have to individually fit single pixels. We can compute a vector of regression coefficients $\vec{a}^* = \left(\cdots, a^*_{i,j,c}, \cdots\right)^T$ simultaneously for the whole image. To avoid overfitting we add a sparsity constraint,

$$\vec{a}^*, b^* = \min_{\vec{a},b} \sum_{t\in\mathcal{T}} \left(\sum_{i,j,c} a_{i,j,c}x_{i,j,c} + b - T(t)\right)^2 \tag{7}$$

$$\text{s.t.} \|a\|_0 = \tau \tag{8}$$

	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)
$R^2(m_x, T)$	0.33 / 8.02	0.38 / 9.29	0.00 / 11.22	0.20 / 8.00	0.00 / 11.12	0.11 / 4.25	0.00 / 10.23	0.08 / 10.29	0.09 / 3.83	0.01 / 11.04
$R^2(m_y, T)$	0.01 / 9.74	0.07 / 11.43	0.00 / 11.21	0.18 / 8.10	0.03 / 10.98	**0.60 / 2.86**	0.17 / 9.31	0.41 / 8.24	0.07 / 3.87	0.12 / 10.40
$R^2(m_x^{lf}, T)$	0.32 / 8.05	**0.62 / 7.25**	0.00 / 11.20	0.37 / 7.11	0.00 / 11.12	0.10 / 4.28	0.00 / 10.23	0.09 / 10.19	0.07 / 3.87	0.03 / 10.91
$R^2(m_y^{lf}, T)$	0.02 / 9.69	0.12 / 11.09	0.01 / 11.17	0.32 / 7.40	0.03 / 10.98	**0.61 / 2.83**	0.24 / 8.94	**0.64 / 6.40**	0.03 / 3.95	**0.47 / 8.06**

Table 2: **Correlating temperatures and motion**: for each scene, we compute camera motion and use its components and their low-frequency versions to regress the temperature. Each cell contains two values, R^2 (coefficient of determination) and Celsius Root Mean Squared Error, (higher R^2 and lower RMSE is better). For some sequences more than half of the variance in the temperature can be "explained" by the motion.

where $\|\cdot\|_0$ counts the number of nonzero elements and τ is a parameter specifying the desired number of nonzero coefficients. We set $\tau = 4$ for all our experiments. We approximate the solution to the optimization problem 7 in two stages. First we use Orthogonal Matching Pursuit [3] (as implemented in SPAMS [11]) to select τ indices i^*, j^*, c^* and then we solve a least squares problem to determine the values of the coefficients $a^*_{i^*,j^*,c^*}$ all other coefficients are set to 0. The prediction is given by $\hat{T}'(t) = a^*_{i,j,c}x'_{i,j,c} + b^*$.

The next two methods use more sophisticated features which are extracted from the whole image.

Convolutional Neural Network (CNN) We compute CNN features using [17] with the deep network $VGG-16$ of Simonyan et al. [15] and train a linear ν-SVR using libsvm [2]. We tune and set the SVR parameters as $C = 100$ and $\nu = 0.5$ for all our experiments. We experimented with the three fully connected (FC) layers, and with the concatenation of all three. The best results, which we report in Table 3 were achieved with features extracted from the first fully connected layer.

Transient Image Attributes (TA) Finally, we train an estimator using the 40 transient attributes of [8]. These attributes try to capture high level transient image properties such as "snow", "autumn" or "fog" which affect scene appearance. Due to the relatively low dimension of this descriptor we opt for a radial basis function ν-SVR. We use $C = 100, \nu = 0.5$ and $g = 1$.

4.1.1 Evaluation and Discussion

We evaluate our predictions using the *coefficient of determination* of Equation (1) and using RMSE as in Equation (2). In Section 3 we interpreted R^2 as a squared correlation coefficient. We can also think of R^2 as the ratio of the Mean Squared Error (MSE) of a predictor and the MSE of the best possible constant predictor (the mean of the temperature). Unlike the setting of Section 3 we are now computing R^2 for test data, thus R^2 can attain negative values (since the bias of the predictor need not match the mean of the test temperature).We consider any predictor with $R^2 < 0$ to be a poor one, and from here on (with a slight abuse of notation) report a nonnegative value $R^2 = \max(0, R^2)$.

The results comparing all the predictions are presented in Table 3. These results are overall encouraging and suggest the possibility of using outdoor webcams as scene-specific temperature sensors. We also note that the performance of the estimators which use simple features is as good, or even better than methods which use more sophisticated global image features.

We believe that part of the success of the single pixel prediction methods can be attributed to the fact that historic measurements from a previous year are pretty good predictors for the current year. During training we score pixels by how well they match the temperature over a whole year. If the test year temperature is similar to that of the training year and the pixel exhibits similar photometric fluctuations during that test year, the prediction should be good.

To quantify this effect we computed R^2 values for the method that predicts today's temperature to be the same as last year's on the same day. We denote this method by "Last Year" (LY) and report the R^2 values in Table 3. We note an important difference between LY and the other methods suggested in the section. In order to look up last year's temperature the algorithm must know the date of the query, whereas the image based algorithms use only image data as the query, no date is needed. It is encouraging to see that predictions based on image content can outperform predictions based on historic measurements.

Another baseline, suggested by one of the anonymous reviewers, is to use the temperature of the nearest training image (using ℓ_2 distance) as the prediction. This baseline which we denote "NN Image" is comparable to the LY baseline and never outperforms the more sophisticated methods. The results are reported in Table 3.

Additional baseline features which we tried include the mean intensity of the image and the means of the different color channels. We also tried using bag-of-visual-words features computed over the sky regions in the images. All of these methods performed poorly.

4.2. Image synthesis

We now consider another way to assess the strength of the correlation between color and temperature, we use temperature as a predictor for image values. Here, the inputs are the regression coefficients $a^*_{i,jc}, b^*_{i,j,c}$ learned with our Local Regression method and a target temperature T. The

	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)
Last Year	0.42 / 9.14	0.56 / 8.16	0.54 / 7.53	0.41 / 5.44	0.61 / 7.35	0.00 / 4.30	0.67 / 6.20	0.59 / 6.77	0.00 / 4.84	0.61 / 7.64
NN Image	0.47 / 8.72	0.59 / 7.83	0.51 / 7.73	0.15 / 6.51	0.13 / 10.92	0.00 / 4.57	0.16 / 9.89	0.70 / 5.83	0.00 / 4.44	0.62 / 7.47
Local Regression	**0.67 / 6.85**	0.65 / 7.24	0.70 / 6.03	0.59 / 4.53	0.76 / 5.77	0.38 / 3.19	0.50 / 7.63	0.77 / 5.09	0.10 / 3.68	0.59 / 7.77
LR Temporal Win.	0.61 / 7.52	0.69 / 6.86	**0.72 / 5.82**	**0.64 / 4.23**	**0.79 / 5.39**	**0.53 / 2.77**	0.54 / 7.35	0.76 / 5.22	0.11 / 3.67	0.58 / 7.85
Global Ridge Reg.	0.00 / 18.16	0.78 / 5.74	0.00 / 35.02	0.00 / 11.37	0.00 / 43.51	0.10 / 3.84	**0.74 / 5.54**	0.00 / 13.86	0.23 / 3.41	0.46 / 8.91
Convolutional NN	0.49 / 8.55	**0.79 / 5.59**	0.71 / 5.96	0.24 / 6.17	0.61 / 7.36	0.48 / 2.90	0.39 / 8.48	**0.79 / 4.88**	**0.43 / 2.93**	0.66 / 7.12
Transient Attrib.	0.36 / 9.60	0.70 / 6.69	0.58 / 7.20	0.55 / 4.75	0.68 / 6.62	0.21 / 3.59	0.58 / 7.03	0.65 / 6.31	0.16 / 3.56	**0.67 / 7.00**

Table 3: **Predicting temperature:** for each scene we train a predictor using data from one year and test it using data from the following year. To evaluate the quality of the prediction we show in each cell two values, R^2 (coefficient of determination) and Celsius Root Mean Squared Error, (higher R^2 and lower RMSE is better).

output is an estimate of the intensity predicted independently for every color channel of every pixel. This can be seen as a very simple way of performing temperature-guided image synthesis. The synthesized image is given by $\hat{x}_{i,j,c} = \frac{1}{a^*_{i,j,c}} \left(T - b^*_{i,j,c}\right)$.

(a) 0 °C 32 °F, (b) 11 °C 52 °F, (c) 26 °C 78 °F,

(d) 0 °C 32 °F, (e) 13 °C 56 °F, (f) 26 °C 78 °F,

Figure 4: Synthesizing scene appearance at different temperatures using LR regression coefficients (denoised with a 3×3 median filter).

Figure 4 shows example synthesis results for two regions of high correlations extracted from the scenes of columns (c) and (e) of Figure 3. We can see that this very simple synthesis method generates visually plausible results. The trees have leaves in the warmer synthesized image, as they do in summer, but do not have them in the synthesized image which corresponds to winter temperatures. The synthesized dome shows a bright sun reflection in the warm synthetic image and a darker shaded dome in the colder image. The artifacts apparant in the synthesized images occur at pixels where the linear regression does not model the relationship between intensity and temperature well. A more sophisticated synthesis application could identify such pixels by their low training R^2 values and denoise or inpaint their values.

5. Discussion

In this paper we explored interactions between the appearance of an outdoor scene and the ambient temperature. Our observations suggest that there are two interesting interactions. First, between intensity measurements at certain regions of the image and the temperature, and second, between the motion of the camera and the temperature. The simple scene-specific temperature prediction algorithms we proposed could turn a camera into a crude temperature sensor. We see this as a first step towards devising a general non-scene specific temperature estimator, with the eventual goal of being able to estimate the temperature from any given outdoor image.

A. Sequence Alignment

As mentioned above, our data preprocessing includes sequence alignment. Even though they are placed on static mounts, most of the cameras in the AMOS datasets exhibit small and occasionally large motion between frames sampled on different days. Alignment of frames from outdoor webcams sampled at much shorter time intervals is discussed in [6]. In this section we discuss our method for robust alignment of sequences whose frames are sampled once every day.

Our alignment problem is challenging because of significant changes that occur even between frames sampled on consecutive days. These include (i) photometric changes due to weather and illumination (e.g. Figure 5(a)); (ii) changes in scene content such as objects entering and leaving scene (e.g. Figure 5 (b)); and (iii) occlusion due to fog or water drops on the lens (e.g. Figure 5(c)). We propose a two stage approach, first we compute a sparse subset of pairwise transformations, which we then denoise to produce a *globally consistent* set of transformations between *all pairs* of images.

Pairwise Alignment We model pairwise transformations using homographies. Our initial attempts to align images using dense intensity based methods and sparse feature based methods were not sufficiently robust due to the significant photometric changes in the sequences. We therefore compute pairwise alignments using the SIFT Flow al-

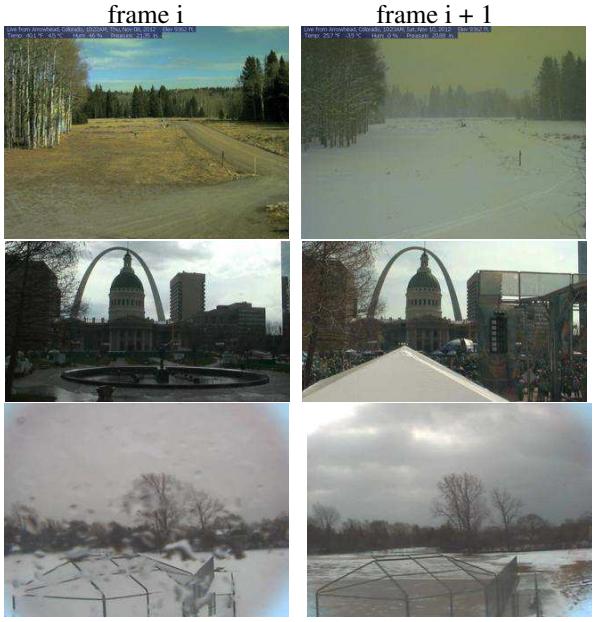


Figure 5: Challenges of aligning consecutive frames. (a) weather and illumination changes (b) objects entering the scene (c) water droplets on the lens.

gorithm [9]. This method computes a dense flow-field and is thus not sensitive to feature matching failures, furthermore it provides us with the photometric invariants afforded by the SIFT descriptor. We use the flow field output by SIFT Flow as putative correspondences and fit a global image homography using *M-estimator Sample Consensus* (MSAC) [16] as implemented in Matlab. The robust MSAC algorithm helps with problem (ii) as it allows us to reject outlier correspondences which can occur due to objects moving and entering or exiting the scene. We compute forward and backwards pairwise alignments for each frame with its 10 closest neighbors as well as with further frames sampled at a density which diminishes like $\frac{1}{t}$ where t is the distance to the current frame.

Denoising and Global Consensus Following the pairwise alignment we are still faced with two challenges. The pairwise method is not perfect and thus some of the computed alignments are wrong of missing. Furthermore, the computed alignments need not be globally consistent. To address these issues we suggest a global denoising method similar to that of Arie-Nachimson et al. [1] and to the methods suggested in [14]. This step allows us to fuse the pairwise estimates to produce a globally consistent and more accurate alignment which results in stabilized sequences. We now briefly explain this process, further details can be found in [1, 14].

Let $\{H_i\}_{i=1}^n$ be the unknown "clean" transformation for image i in some global coordinate system. Assuming all hompographies are normalized to have determinant 1, $H \leftarrow \det(H)^{-\frac{1}{3}} H$, the clean pairwise transformation $H_{i,j}$, should obey the pairwise relationship $H_{i,j} = H_j H_i^{-1}$.

Given the (normalized) noisy estimates $\hat{H}_{i,j}$ from the pairwise alignment step we would like to find clean transformations such that

$$\min_{H_1,\cdots,H_n} \sum_{i,j} \|H_j H_i^{-1} - \hat{H}_{i,j}\|_F^2. \tag{9}$$

Let $U = (H_1, \cdots, H_n)^T$, $V = \left(H_1^{-1}, \cdots, H_n^{-1}\right)^T$ and $G = UV^T$. Note that $GU = nU$, that is, the three columns of U are the leading eigenvectors of G with eigenvalue n.

We can approximate the solution to problem (9) by computing the 3 leading eigenvectors of the matrix

$$\hat{G} = \begin{pmatrix} I_{3\times 3} & \cdots & \hat{H}_{n,1} \\ \vdots & \ddots & \vdots \\ \hat{H}_{1,n} & \cdots & I_{3\times 3} \end{pmatrix}. \tag{10}$$

To handle failures and missing entries in our pairwise alignment we zero the corresponding blocks of $\hat{G}$ and scale its rows by dividing with the number of valid transformations for that row. We use $\hat{U}$, the three leading eigenvectors of $\hat{G}$, to construct our globally consistent estimates. Numerical errors may lead to complex estimates in which case we take the real part of the estimate.

Results Using the consistent pairwise estimates we generate a stabilized sequence by transforming all frames to a single reference frame. In Figure 6 we can see the results of the alignment demonstrated on one of the sequences. In 6a we show the mean of the aligned sequence compared to the mean of the non-aligned sequence. In 6b we show the difference between the first and last frame of the aligned vs. the non-aligned sequences.

(a) Sequence mean frame. (b) abs(first - last) frame

Figure 6: Sequence alignment results.

Acknowledgments: DG and TZ acknowledge support from NSF award IIS-1212928. The research of LZM is supported by the Israel Science Foundation under grant 1179/11.

References

[1] M. Arie-Nachimson, S. Z. Kovalsky, I. Kemelmacher-Shlizerman, A. Singer, and R. Basri. Global motion estimation from point matches. In *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012 Second International Conference on*, pages 81–88. IEEE, 2012. 8

[2] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm. 6

[3] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM journal on scientific computing*, 20(1):33–61, 1998. 6

[4] M. Islam, N. Jacobs, H. Wu, and R. Souvenir. Images+Weather: Collection, Validation, and Refinement. In *IEEE Conference on Computer Vision and Pattern Recognition Workshop on Ground Truth (CVPRW)*, 2013. 2

[5] N. Jacobs, W. Burgin, N. Fridrich, A. Abrams, K. Miskell, B. H. Braswell, A. D. Richardson, and R. Pless. The global network of outdoor webcams: properties and applications. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 111–120. ACM, 2009. 2

[6] N. Jacobs, W. Burgin, R. Speyer, D. Ross, and R. Pless. Adventures in archiving and using three years of webcam images. In *Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009. IEEE Computer Society Conference on*, pages 39–46. IEEE, 2009. 2, 7

[7] N. Jacobs, N. Roman, and R. Pless. Consistent temporal variations in many outdoor scenes. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–6. IEEE, 2007. 2

[8] P.-Y. Laffont, Z. Ren, X. Tao, C. Qian, and J. Hays. Transient attributes for high-level understanding and editing of outdoor scenes. *ACM Transactions on Graphics (proceedings of SIGGRAPH)*, 33(4), 2014. 2, 6

[9] C. Liu, J. Yuen, and A. Torralba. Sift flow: Dense correspondence across scenes and its applications. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(5):978–994, 2011. 8

[10] C. Lu, D. Lin, J. Jia, and C.-K. Tang. Two-class weather classification. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 3718–3725. IEEE, 2014. 2

[11] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online dictionary learning for sparse coding. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 689–696. ACM, 2009. 6

[12] S. Narasimhan and S. Nayar. Vision and the Atmosphere. *International Journal on Computer Vision*, 48(3):233–254, Jul 2002. 2

[13] S. Narasimhan, C. Wang, and S. Nayar. All the Images of an Outdoor Scene. In *European Conference on Computer Vision (ECCV)*, volume III, pages 148–162, May 2002. 2

[14] P. Schroeder, A. Bartoli, P. Georgel, and N. Navab. Closed-form solutions to multiple-view homography estimation. In *Applications of Computer Vision (WACV), 2011 IEEE Workshop on*, pages 650–657. IEEE, 2011. 8

[15] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 6

[16] P. H. Torr and A. Zisserman. MLESAC: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78(1):138–156, 2000. 8

[17] A. Vedaldi and K. Lenc. Matconvnet – convolutional neural networks for matlab. *CoRR*, abs/1412.4564, 2014. 6